

Final Presentation

교내 장학금 추천 서비스

Team 10 권준 김동현 김지훈 박지웅 박진아 오승재 이정민



SKKU
장학비서

목차 Contents

- 1. 프로젝트 동기 및 overview
- 2. Implement
- 3. Demo Video
- 4. Test
- 5. Conclusion



SKKU
장학비서

프로젝트 동기

분산된
장학 정보

학과 및 단과대 홈페이지,
학교 공식 사이트

높은 검색
피로도

자신이 받을 수 있는
장학금 공지를 찾는 어려움

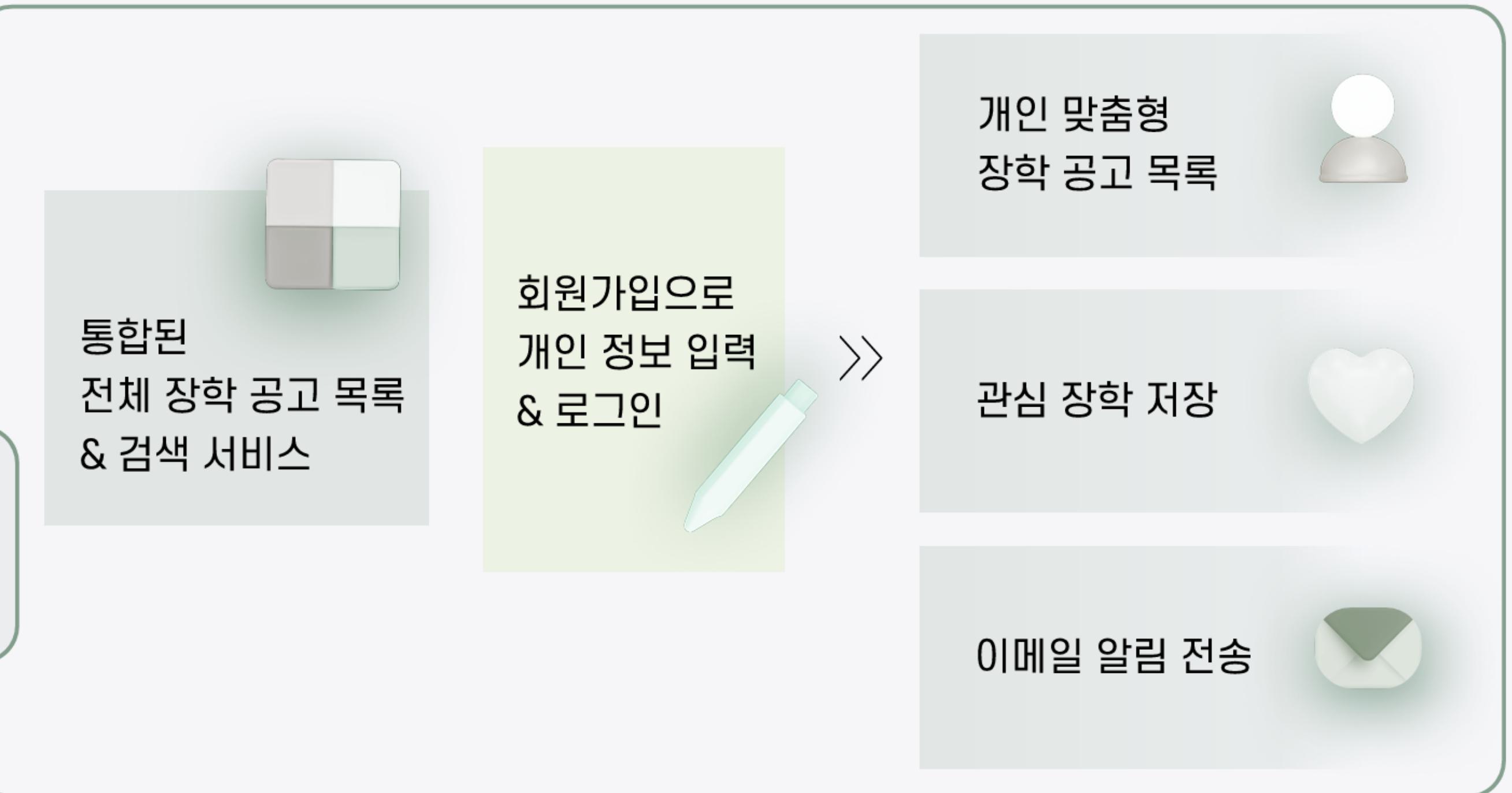
현 시스템의
기능 부재

챌린지스퀘어에서
제공하지 않는
장학 관련 서비스

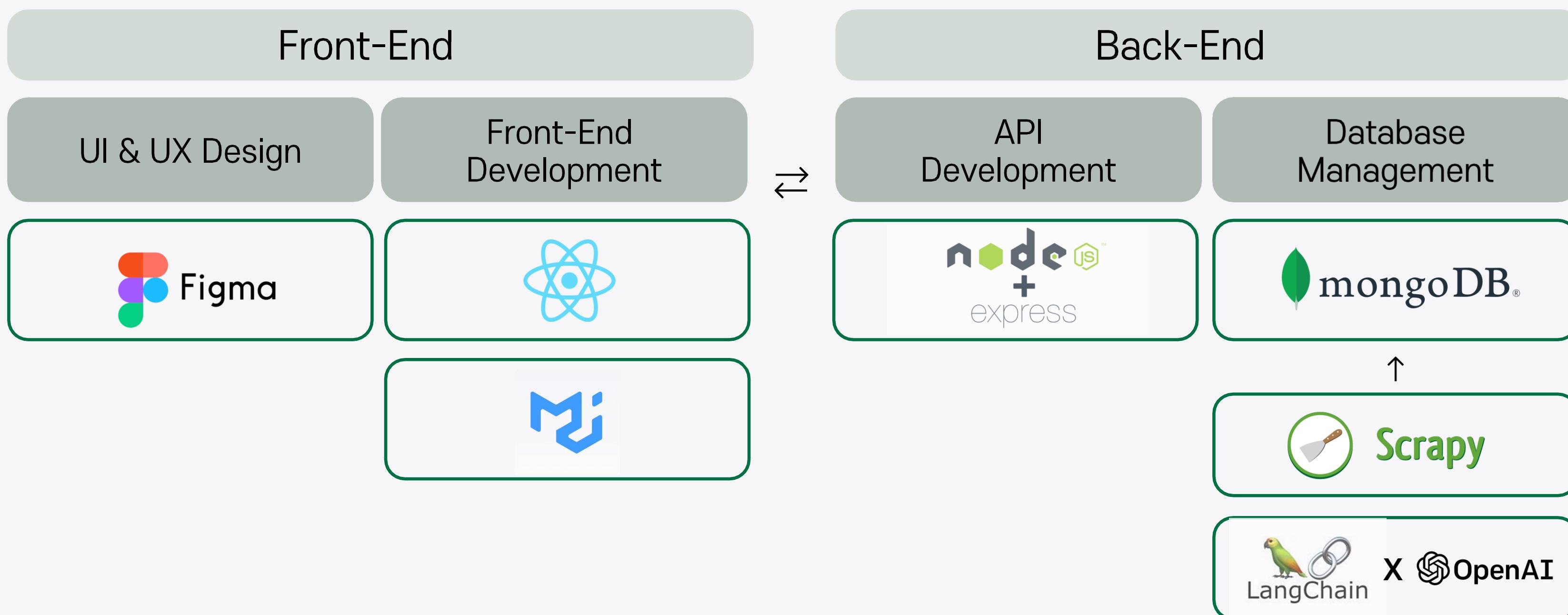


맞춤형 장학 추천
& 알림 서비스

서비스 overview



Tech Stack



회원가입 가능

Implement - Front

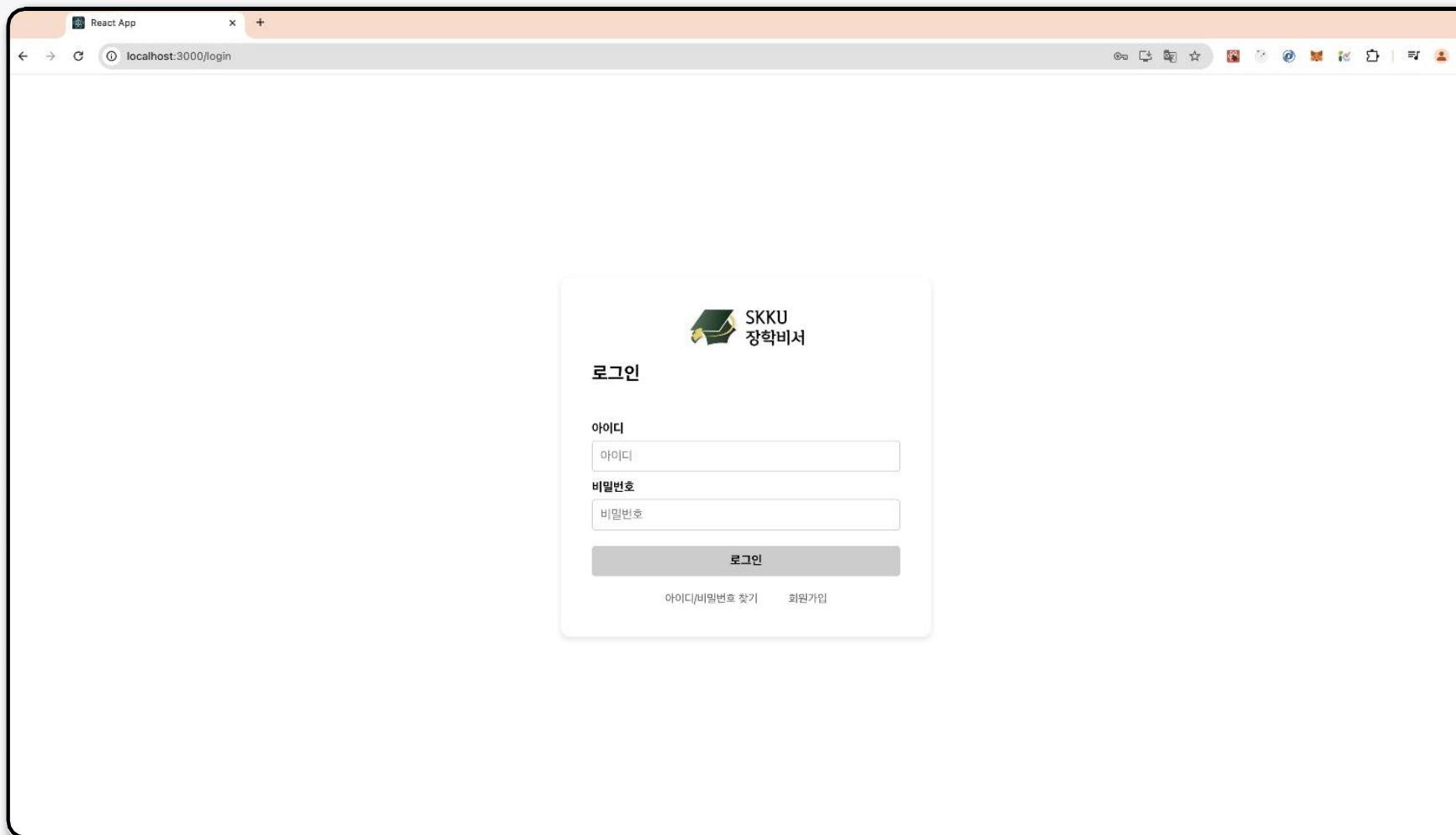
스꾸장학비서의 개인화된 서비스를 이용하기 위해 사용자 회원가입을 제공
회원가입은 총 3단계로 구성

The figure displays three sequential screenshots of a web-based user registration form for 'SKKU 장학비서'.
1. The first screenshot shows the initial registration page with fields for '아이디' (ID), '비밀번호' (Password), and '비밀번호 확인' (Password Confirmation).
2. The second screenshot shows the second step of the registration process, titled '기본 정보 입력' (Basic Information Input), which includes fields for '성명' (Name), '생년월일' (Birthdate), '이메일' (Email), '전공' (Major), '현재 학기' (Current Semester), and three dropdown menus for '선택' (Select), '중학' (Middle School), and '수료' (Graduated).
3. The third screenshot shows the final step, titled '추가 정보 입력' (Additional Information Input), which includes fields for '전체 학기 평점' (Overall Grade Point Average), '직전 학기 평점' (Previous Grade Point Average), '소득 분위' (Income Decile), and '거주지' (Residence). A large '다음' (Next) button is visible at the bottom.

- 사용자에게 맞춤형 장학공고를 추천하기 위해 아이디와 비밀번호, 전공, 평점, 소득 분위 등 관련 정보를 입력받음
- 회원가입 완료시 로그인 화면으로 연결

로그인 기능

Implement - Front



- 회원가입시 설정한 아이디와 비밀번호로 로그인하는 기능
- 로그인 성공 시 메인화면으로 연결

+ 부가기능

아이디/비밀번호 찾기 버튼을 통해
계정 복구가 필요한 회원이 비밀번호를
재설정할 수 있도록 해당 화면으로 연결

회원 정보 조회 및 수정 기능

Implement - Front

The image displays two screenshots of a React application interface for member information management. Both screenshots show a dark-themed dashboard with various user profiles and search results.

Left Screenshot (회원 정보 조회): This screen shows a list of member profiles. One profile is highlighted with a light gray background, showing details such as:

- Profile picture: jin
- Name: jin
- Gender: 남자
- Current Grade: 8학기
- Academic Status: 재학
- Major: 소프트웨어학과
- GPA: 3.9 / 4.5
- Attendance: 4.1 / 4.5
- Residence: 서울 용산구

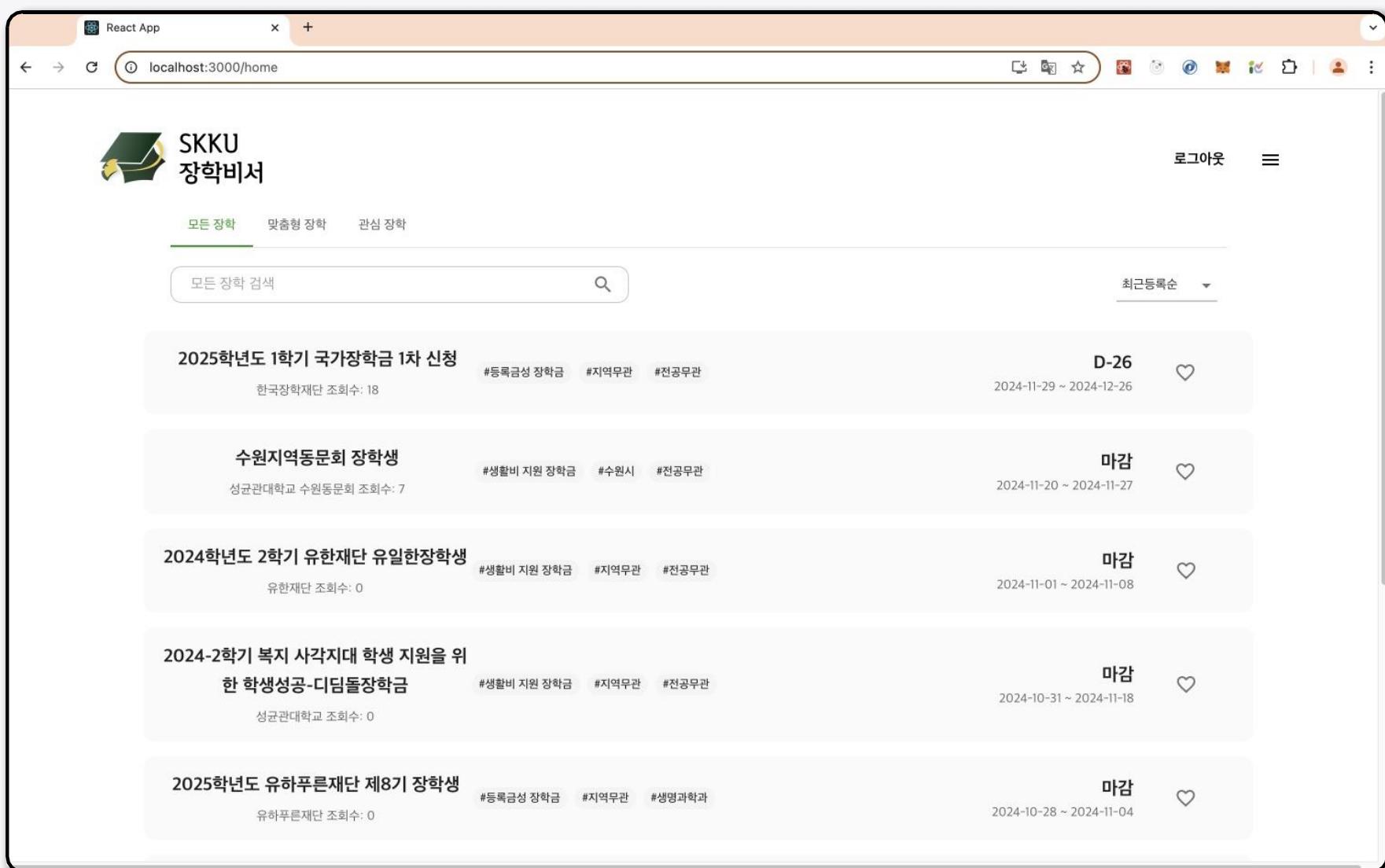
Right Screenshot (회원 정보 수정): This screen shows the same profile with edit fields overlaid. The fields include:

- Current Grade: 9학기
- Academic Status: 재학
- GPA: 4.0
- Attendance: 4.3
- Residence: 서울 용산구

메인 화면의 우측 상단 햄버거 아이콘을 클릭 시 우측에서 사이드바가 나타남
회원가입 시 입력한 현재 학기, 학적 상태, 평점 등 회원 정보를 확인하고 수정 가능

전체 장학 공고 목록 화면

Implement - Front



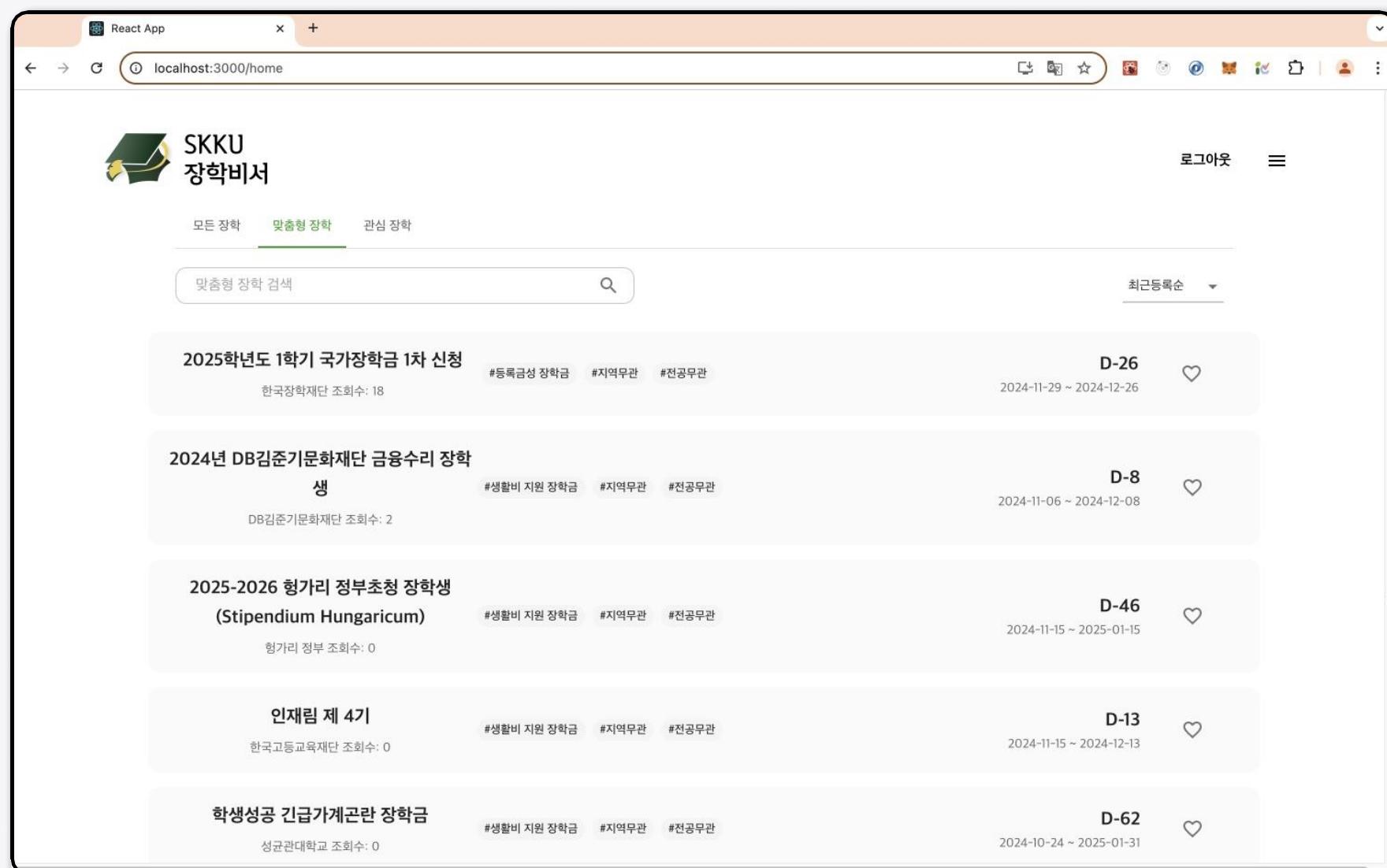
- 메인 화면의 첫 번째 탭의 ‘모든 장학’ 클릭 시 표시되는 화면
- 각 단과대 및 학교 전체 공지사항에 올라오는 장학 정보를 한 곳에 모두 모아 표시
- 비회원/회원 모두 열람 가능

+ 부가 기능

- 공고 클릭 시 해당 공고의 상세 화면으로 이동
- 하트 클릭 시 관심 공고 리스트에 추가/제거
- 검색창을 이용해 장학 공고 검색 가능

맞춤형 장학 기능

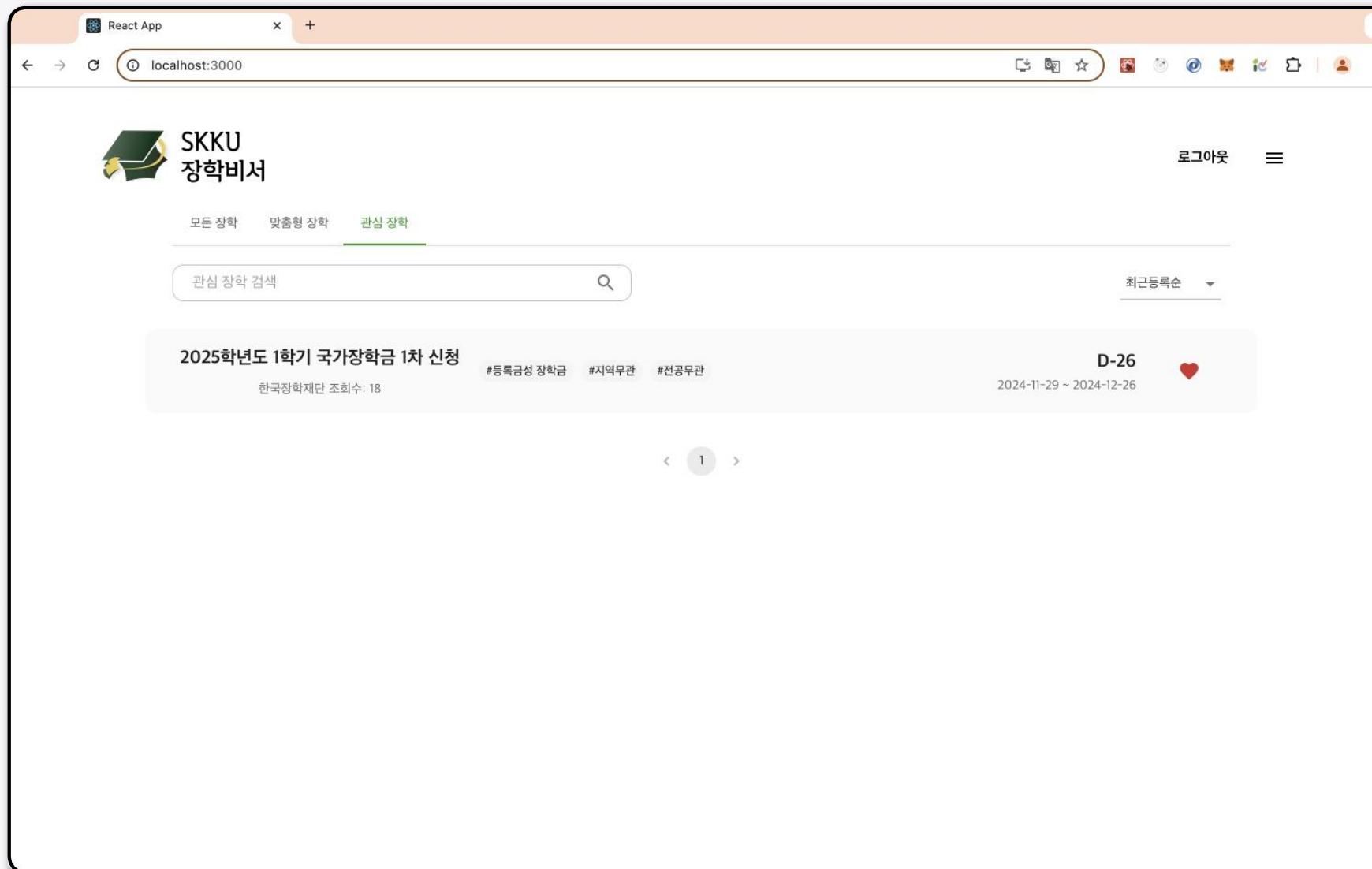
Implement - Front



- 메인 화면의 두 번째 탭의 '맞춤형 장학' 클릭 시 표시되는 화면
- 전체 장학 중 회원 정보를 기반으로 지원 자격을 검토한 후 개인이 지원 가능한 장학 공고만을 선별해 제공
- 회원에게만 제공되는 기능으로, 비회원 사용자가 맞춤 장학 탭을 클릭하면 로그인이 필요한 서비스라고 안내하며 로그인 화면으로 이동

관심 장학 기능

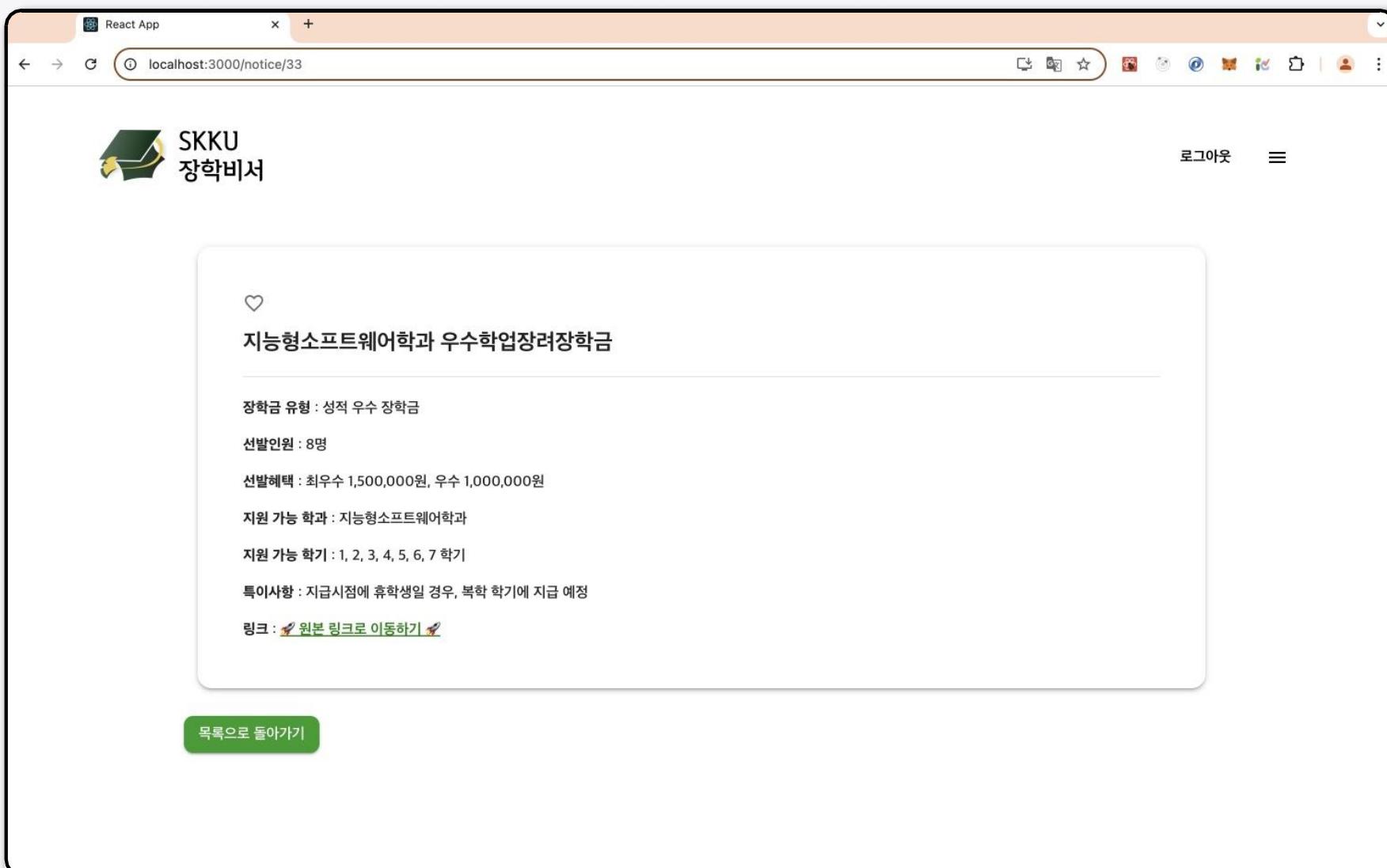
Implement - Front



- 메인 화면의 세 번째 탭의 '관심 장학' 클릭 시 표시되는 화면
- 사용자가 관심 등록을 한 공고만 제공
- 회원에게만 제공되는 기능으로, 비회원 사용자가 관심 장학 탭을 클릭하면 로그인이 필요한 서비스라고 안내하며 로그인 화면으로 이동

장학 공고 상세 화면

Implement - Front



- 특정 장학 공고를 클릭하면 표시되는 장학 공고 상세 화면
- 지원 자격 등 중요한 정보를 중심으로 장학금의 상세 내용을 제공하며, 추가 정보를 쉽게 확인할 수 있도록 장학 공고 원본 사이트로의 쿼링크도 제공
- 상세 화면에서 '목록으로 돌아가기' 버튼을 클릭하면 다시 메인 화면으로 돌아감

Database

Implement - Back

Scholarship

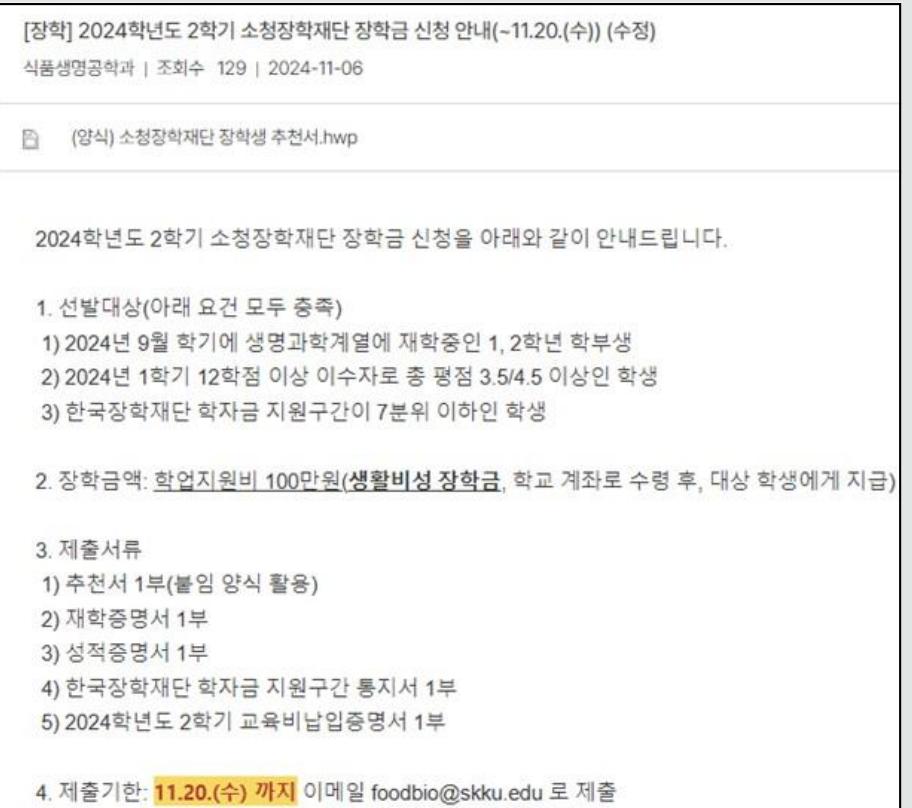
Field	Type	Constraint	Description
_id	Number	PK, NOT NULL	장학금 고유 번호
scholarshipName	String	NOT NULL	장학금 이름
eligibleMajor	Array (String)		지원 대상 학과
minimumGPARequirement	Number		최소 학점 기준
compTotalGPA	Boolean		총 학점 기준 여부
eligibleSemesters	Array (Number)		지원 가능 학기
scholarshipType	String	NOT NULL	장학금 유형
ageLimit	Number		나이 제한
regionalRestrictions	Array (String)		지역 제한
incomeLevelRequirement	Number		소득분위 기준
applicationPeriod	String	NOT NULL	신청 기간
scholarshipAmount	String		장학금 금액
numberOfRecipients	Number		선발 인원
requiredDocuments	Array (String)		제출 서류
applicationMethod	String		신청 방법
significant	String		특이사항
link	String	NOT NULL	링크
views	Number	NOT NULL	조회수
foundation	String	NOT NULL	재단 이름
uploadedDate	String	NOT NULL	DB에 추가된 날짜
lastUploadedDate	String		이전 uploadedDate

User

Field	Type	Constraint	Description
_id	Number	PK, NOT NULL	유저 고유 번호
userID	String	NOT NULL, Unique	유저 ID
userPassword	String	NOT NULL	유저 비밀번호
userEmail	String	NOT NULL, Unique	유저 이메일
gender	Boolean	NOT NULL	성별
birthday	Date	NOT NULL	생년월일
currentSemester	Number	NOT NULL	현재 학기
currentStatus	String	NOT NULL	학적 상태
major	String	NOT NULL	원 전공 학과
incomeLevel	Number		소득 분위
lastGPA	Number		작전 학기 평점
totalGPA	Number		총 평균 평점
region	String		거주지
savedScholarship	Array (Integer)		관심 장학

Data Crawling

Implement - Back



Scrapy

Scrapy library 활용

대상 : 학교 공식 홈페이지 및 52개 학과별 홈페이지

결과 : 공지사항을 크롤링하여 장학금 데이터 생성

```
skku_notice > {} notices.json > ...
1  [
2    {
3      "title": "2024학년도 2학기 소청장학재단 장학금 신청 안내(~11.20.(수))",
4      "link": "https://skb.skku.edu/foodlife/community/notice.do?mode=view&no=129",
5      "department": "식품생명공학과",
6      "start_date": "2024-11-06",
7      "content": "2024학년도 2학기 소청장학재단 장학금 신청을 아래와 같이 안내드립니다."
8    }
9  ]
```

Data Parsing

Implement - Back

```
_id: 138
scholarshipName : "2024학년도 2학기 소청장학재단 장학금"
▼ eligibleMajors : Array (4)
  0: "생명과학과"
  1: "식품생명공학과"
  2: "바이오메카트로닉스학과"
  3: "융합생명공학과"
minimumGPARequirement : 3.5
compTotalGPA : true
▼ eligibleSemesters : Array (4)
  0: 1
  1: 2
  2: 3
  3: 4
scholarshipType : "생활비 지원 장학금"
ageLimit : null
regionalRestrictions : null
incomeLevelRequirement : 7
applicationPeriod : "2024-11-06 ~ 2024-11-20"
scholarshipAmount : "학업지원비 100만원"
numberOfRecipients : null
▼ requiredDocuments : Array (5)
  0: "추천서"
  1: "재학증명서"
  2: "성적증명서"
  3: "한국장학재단 학자금 지원구간 통지서"
  4: "교육비납입증명서"
applicationMethod : "이메일 제출 (foodbio@skku.edu)"
significant : null
link : "https://skb.skku.edu/foodlife/community/notice.do?mode=view&articleNo=..."
views : 0
foundation : "소청장학재단"
uploadedDate : "2024-11-22"
lastUploadedDate : null
```



LangChain framework & GPT-4 model of OpenAI 활용

Model Chaining

- First Model (gpt-4-32k): 공지사항 내용을 분류하고, 지정된 형식에 맞게 파싱(parsing)
- Second Model (gpt-4-8k): 지역 및 학과 정보 등 세부정보를 업데이트하여 데이터 정제(refinement)

Chunking

- 데이터를 작은 chunk로 나누어 모델 프롬프트에 추가
- 토큰 수 제한으로 인한 대용량 데이터의 정확도 저하 방지

MongoDB Atlas와 연결하여 parsing된 데이터 삽입

Data Parsing Result

Implement - Back

원문

[장학] 2024학년도 2학기 소청장학재단 장학금 신청 안내(~11.20.(수)) (수정)
식품생명공학과 | 조회수 129 | 2024-11-06

(양식) 소청장학재단 장학생 추천서.hwp

2024학년도 2학기 소청장학재단 장학금 신청을 아래와 같이 안내드립니다.

1. 선별대상(아래 요건 모두 충족)

- 1) 2024년 9월 학기에 생명과학계열에 재학중인 1,2학년 학부생
- 2) 2024년 1학기 12학점 이상 이수자로 총 평점 3.5/4.5 이상인 학생
- 3) 한국장학재단 학자금 지원구간이 7분위 이하인 학생

2. 장학금 액: 학업지원비 100만원(생활비성 장학금, 학교 계좌로 수령 후, 대상 학생에게 지급)

3. 제출서류

- 1) 추천서 1부(불임 양식 활용)
- 2) 재학증명서 1부
- 3) 성적증명서 1부
- 4) 한국장학재단 학자금 지원구간 통지서 1부
- 5) 2024학년도 2학기 교육비납입증명서 1부

4. 제출기한: 11.20.(수) 까지 이메일 foodbio@skku.edu 로 제출

결과

```
-id: 138
scholarshipName : "2024학년도 2학기 소청장학재단 장학금"
▼ eligibleMajors : Array (4)
  0: "생명과학과"
  1: "식품생명공학과"
  2: "바이오메카트로닉스학과"
  3: "융합생명공학과"
minimumGPARequirement : 3.5
compTotalGPA : true
▼ eligibleSemesters : Array (4)
  0: 1
  1: 2
  2: 3
  3: 4
scholarshipType : "생활비 지원 장학금"
ageLimit : null
regionalRestrictions : null
incomeLevelRequirement : 7
applicationPeriod : "2024-11-06 ~ 2024-11-20"
scholarshipAmount : "학업지원비 100만원"
numberOfRecipients : null
▼ requiredDocuments : Array (5)
  0: "추천서"
  1: "재학증명서"
  2: "성적증명서"
  3: "한국장학재단 학자금 지원구간 통지서"
  4: "교육비납입증명서"
applicationMethod : "이메일 제출 (foodbio@skku.edu)"
significant : null
link : "https://skb.skku.edu/foodlife/community/notice.do?mode=view&articleNo=..."
views : 0
foundation : "소청장학재단"
uploadedDate : "2024-11-22"
lastUploadedDate : null
```

Data Filtering

Implement - Back

전공, 학점, 학기(학년), 나이, 지역, 소득분위, 지원기간 별로
장학금 정보와 사용자의 정보를 비교하여 적합하지 않을 경우 추천 장학금에서 제외

```
function buildMatchingScholarships(user) {
  return function (scholarship) {
    // eligibleMajors
    if (
      scholarship.eligibleMajors !== null &&
      !scholarship.eligibleMajors.includes(user.major)
    ) {
      return false;
    }

    // minimumGPARequirement
    if (scholarship.minimumGPARequirement !== null) {
      const gpaToCompare = scholarship.compTotalGPA ? user.totalGPA : user.lastGPA;
      if (scholarship.minimumGPARequirement > gpaToCompare) {
        return false;
      }
    }

    // eligibleSemesters
    if (
      scholarship.eligibleSemesters !== null &&
      !scholarship.eligibleSemesters.includes(user.currentSemester)
    ) {
      return false;
    }

    const today = new Date();
    const userBirthDay = user.birthday;
    const userAge = today.getFullYear() - userBirthDay.getFullYear();
    // ageLimit
    if (scholarship.ageLimit !== null && userAge > scholarship.ageLimit) {
      return false;
    }
  }
}
```

지원 가능 전공제약이 있으면
사용자의 전공이 지원가능 전공인지 확인

학점 제한이 있으면, 기준 학기 확인 후
사용자의 학점과 비교

소득분위 제한이 존재하면
사용자의 소득분위와 비교

학기 혹은 학년 제한이 있으면
사용자의 현재 학기가 지원가능 학기인지 확인

나이제한이 있으면
사용자의 생일과 날짜를 비교하여 나이를 계산한 후,
지원 가능한 나이인지 확인

```
// regionalRestrictions
if (
  scholarship.regionalRestrictions !== null &&
  !scholarship.regionalRestrictions.includes(user.region)
) {
  return false;
}

// incomeLevelRequirement
if (
  scholarship.incomeLevelRequirement !== null &&
  user.incomeLevel > scholarship.incomeLevelRequirement
) {
  return false;
}

if (!scholarship.applicationPeriod) return false; // Exclude if no applicationPeriod
const [start, end] = scholarship.applicationPeriod.split('~').map(date => new Date(date.trim()))
return today >= start && today <= end; // Check if today is within the range
}
```

지역 제한이 있으면
사용자의 지역과 비교

지원 기간이 현재 일자에
해당하는지 확인

Data Filtering

Implement - Back

유저에게 보내지는 장학금 데이터 구조

```
1
2
3  function compactScholarship(scholarship, savedScholarshipIDs = []) {
4    const scholarshipID = scholarship._id;
5    return {
6      scholarshipID,
7      scholarshipName: scholarship.scholarshipName || "Unknown Scholarship",
8      applicationPeriod: scholarship.applicationPeriod || "Unknown Period",
9      foundation: scholarship.foundation || "Unknown Foundation",
10     views: scholarship.views || 0,
11     isFav: savedScholarshipIDs.includes(scholarshipID),
12     tags: scholarship.tags || [
13       scholarship.scholarshipType || "General",
14       scholarship.regionalRestrictions?.[0] || "지역무관",
15       scholarship.eligibleMajors?.[0] || "전공무관"
16     ]
17   };
18 }
19
20 module.exports = compactScholarship;
```

개별 장학 조회가 아닌 다수 장학 조회 상황에서,
데이터 베이스에 저장된 장학금 전체 정보는
유저가 한 번에 파악하기 어렵다

따라서 데이터를 가공 후 전달

Tags를 통해 장학금에 대한 대략적인 정보 확인 가능

Email system

Implement - Back

1 최근 업데이트된
관심 설정한 장학금,
필터링을 통한 추천 장학금
전부 나열하여 전송
3일 주기로 자동실행

2

skkuscholarship@gmail.com
나에게 ▾
인증 번호를 입력하십시오.
인증 번호:
4lh1bL

비밀번호 재설정을 위한
인증번호 전송
(인증번호는 알파벳, 숫자 조합 6자리)

3

skkuscholarship@gmail.com
나에게 ▾
There is updated information regarding the saved scholarship.

For more information, please refer to the site below.
link: 알 수 없음
장학금: 2024학년도 유한재단 결손/다문화가정 대상 장학생
장학재단: 유한재단
장학종류: 생활비 지원 장학금
선발인원: 2
장학혜택: 1인당 3,000,000원의 생활지원금
신청 기간: 2024-11-14 ~ 2024-11-27
선발 대상: 전공 무관

마감일까지 3일 이하 남은
관심장학 체크하여 전송
매일 자동실행

Email system

Implement - Back

1. nodemailer를 활용한 이메일을 보내는 함수를 구현
2. email에 대한 양식을 각 상황에 따라 나누어서 처리

```
const getEmailContent = (type, data) => {
  switch(type) {
    case 'verification':
      return {
        subject: 'SKKU Scholarship Verification Code',
        text: `Please enter your verification code.
Verification Code:
${data.verifyCode}
.split('\n')
.map(line => line.trim())
.join('\n')
`,
      };
    case 'sendMatchingScholarships':
      subject = 'Updated Info on Recommended Scholarships';
      commonMessage = 'There is updated information regarding the recommended scholarships.
';
    case 'sendSavedScholarshipsBeforeDeadline':
      subject = 'The deadline for your saved scholarship is approaching';
      commonMessage = 'There are 3 days left until the deadline for the saved scholarship.
';
    case 'sendUpdatedSavedScholarships':
      subject = 'Updated Info for your saved Scholarships';
      commonMessage = 'There is updated information regarding the saved scholarships.
';
    default:
      return {
        subject: subject,
        text: commonMessage + getScholarshipsContent(data)
      };
  }
}
```

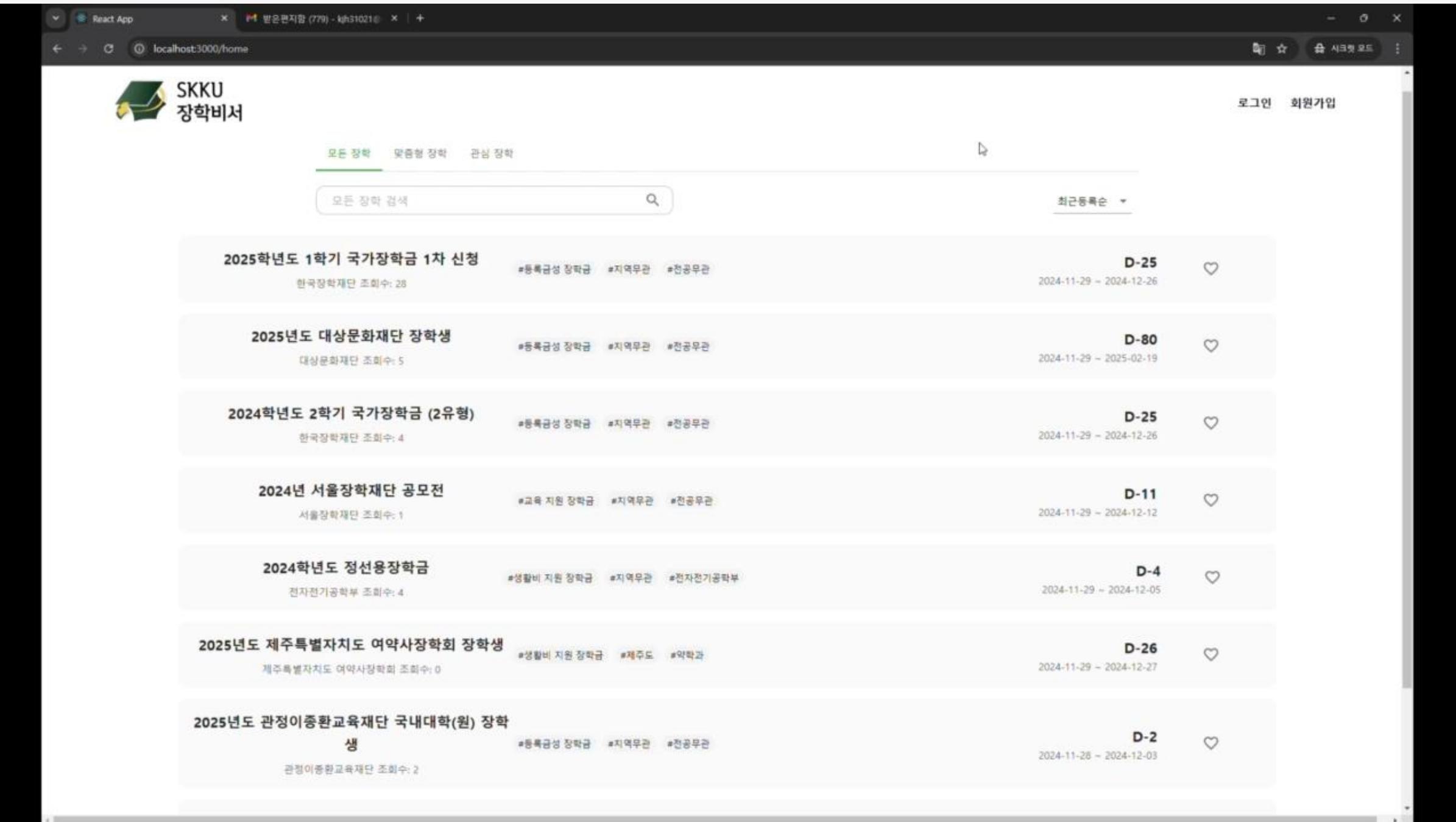
인증 코드
관련 메일 양식

장학금 정보
관련 메일 양식

```
const getScholarshipsContent = (data) => {
  return (
    data.map(scholarship =>
      `For more information, please refer to the site below.
link: ${data.link} || "알 수 없음"
장학금: ${scholarship.scholarshipName} || "알 수 없음"
장학재단: ${scholarship.foundation} || "알 수 없음"
장학종류: ${scholarship.scholarshipType} || "알 수 없음"
선발인원: ${scholarship.numberOfRecipients} || "알 수 없음"
장학혜택: ${scholarship.scholarshipAmount} || "알 수 없음"
신청 기간: ${scholarship.applicationPeriod} || "알 수 없음"
선발 대상: ${!(scholarship.eligibleMajors || []).length > 0
? scholarship.eligibleMajors.map(major => major).join(", ")
: "전공 무관"
}
.split('\n')
.map(line => line.trim())
.join('\n'))
.join('
-----
')
)
}
```

시연 연상

Demo Video



TEST Front-End

Use-Case	Test
회원가입	<ul style="list-style-type: none">이미 존재하는 아이디에 대해 회원가입을 시도할 때 중복확인이 제 기능을 하는지 확인비밀번호를 비밀번호 확인란과 다르게 입력할 때 비밀번호 확인이 제 기능을 하는지 확인기본 정보 입력 시 정보를 다 기입하지 않았을 때 다음버튼이 활성화되지 않는지 확인전공 입력 시 드롭다운에서 선택 가능하고 검색 필터가 제대로 작동하는지 확인학기 평점 입력할 때 0~4.5 사이로만 입력 가능한지 확인
로그인	<ul style="list-style-type: none">존재하는 아이디와 패스워드로 로그인할 때 제대로 로그인 되는지 확인존재하지 않는 아이디로 로그인할 때 잘못된 아이디라고 명시 되는지 확인틀린 패스워드로 로그인할 때 잘못된 패스워드라고 명시 되는지 확인

TEST Front-End

Use-Case	로그인 상태	Test
공고 목록 확인	로그인 여부와 무관	<ul style="list-style-type: none">사이트 접속했을 때 전체 공고가 나오는지 확인사용자가 검색/필터를 수정하였을 때 공고가 정렬되는지 확인
	비로그인	<ul style="list-style-type: none">추천/관심 공고 페이지에 접속할 때 ‘로그인 필요 여부’ 이미지가 뜨는지 확인관심 여부 버튼이 비활성화 되는지 확인
	로그인	<ul style="list-style-type: none">사용자가 검색/필터를 수정하였을 때 공고가 정렬되는지 확인추천/관심 장학 정보가 없을 때 설정한 이미지가 뜨는지 확인
상세 공고 확인	로그인 여부와 무관	<ul style="list-style-type: none">장학 공고 클릭했을 때 상세 정보가 정상적으로 나오는지 확인‘원본 링크로 이동’ 클릭했을 때 해당 공고의 웹사이트로 연결되는지 확인
	비로그인	<ul style="list-style-type: none">관심 여부 버튼이 비활성화 되는지 확인
	로그인	<ul style="list-style-type: none">관심 여부 버튼을 클릭했을 때 반영이 되는지 확인
회원 정보 확인	비로그인	<ul style="list-style-type: none">사이드바 버튼이 사라지는지 확인
	로그인	<ul style="list-style-type: none">사이드바 버튼을 클릭했을 때 회원 정보가 뜨는지 확인

TEST Back-End

- Test Tool: Jest
- Unit Test: 함수와 로직 검증
- Mocking: 외부 의존성을 대체하여 독립적인 테스트 수행



실제 API 요청을 가정한 인증 테스트

```
it('should return 404 if password is incorrect', async () => {
  // Mock User.findOne to return a user
  User.findOne.mockResolvedValue({
    userID: 'example1',
    userPassword: '1398',
  });

  const response = await request(app).post('/').send({
    userID: 'example1',
    userPassword: '1234',
  });

  expect(response.status).toBe(404);
  expect(response.body).toEqual(createResponse(false, "Wrong password", null));
});
```

Mock 데이터 설정

: 데이터베이스 호출을 Mocking하여 독립적으로 인증 로직 검증

테스트 요청 보내기

: 잘못된 비밀번호 입력 시 서버가 적절히 응답하는지 확인

결과 검증

: 404 상태 코드와 오류 메시지가 올바르게 반환되는지 확인

TEST Back-End

- CI Pipeline: GitHub Actions 활용
- Trigger:

PR 생성 시 > main branch로 PR이 생성될 때 테스트 자동 실행

PR 업데이트 시 > PR에 commit이 추가되거나 수정되면 다시 실행

- 자동 테스트 수행:

모든 Unit Test와 Integration Test 실행

```
1  name: CI - Jest Tests
2
3  on:
4    pull_request:
5      branches:
6        - main
```

.github/workflows/test.yml 일부

Feat: add CI configuration for Jest tests (#34)

CI - Jest Tests #10: Commit [64b9ef2](#) pushed by JihunSKKU

main

PR 생성 후 GitHub Actions 테스트 결과

TEST Back-End

- 사용 code test coverage: 100%
모든 함수, 문장, 라인 테스트 완료
- 예외 처리 및 모든 branch coverage: 100%
(총 167개 branch test 완료)

```
===== Coverage summary =====
Statements : 100% ( 440/440 )
Branches   : 100% ( 167/167 )
Functions   : 100% ( 45/45 )
Lines       : 100% ( 429/429 )
=====
Test Suites: 13 passed, 13 total
Tests:      117 passed, 117 total
Snapshots:  0 total
Time:       31.304 s
```

File	%Stmts	%Branch	%Funcs	%Lines	Uncovered Line #s
All files	100	100	100	100	
daemon	100	100	100	100	
sendEmail.js	100	100	100	100	
verificationCode.js	100	100	100	100	
models	100	100	100	100	
CompactScholarship.js	100	100	100	100	
Scholarship.js	100	100	100	100	
User.js	100	100	100	100	
routes	100	100	100	100	
login.js	100	100	100	100	
register.js	100	100	100	100	
scholarships.js	100	100	100	100	
users.js	100	100	100	100	
services	100	100	100	100	
sendMatchingScholarships.js	100	100	100	100	
sendSavedScholarshipsBeforeDeadline.js	100	100	100	100	
sendUpdatedSavedScholarships.js	100	100	100	100	
utils	100	100	100	100	
buildMatchingScholarships.js	100	100	100	100	
compactScholarship.js	100	100	100	100	
responseListTemplate.js	100	100	100	100	
responseTemplate.js	100	100	100	100	

```
Test Suites: 13 passed, 13 total
Tests:      117 passed, 117 total
Snapshots:  0 total
Time:       32.515 s, estimated 33 s
```

Conclusion

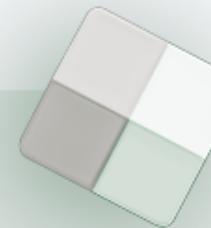


SKKU
장학비서



통합된 장학 정보

학과 및 단과대 홈페이지,
학교 공식 사이트에 분산된
장학 정보를 웹사이트 하나
로 통합한 환경 제공



개인 맞춤형 장학 서비스

자신이 받을 수 있는
장학금 공지 추천으로
일일이 확인하는 번거로움
없이 쉽게 사용 가능



편리한 기타 장학 관련 서비스

기존에 없던 관심 장학,
알림 메일 서비스로
편의성 증대

“ 편리하고 개별화된 장학 지원 프로세스 제공 ”



SKKU
장학비서

Q&A

