

1
2
3
4 // Welcome to the Loop;

5
6
7
8
9 {조 이름: 루프 ;
10
11
12 웹사이트 만들기
13
14
15
16
17 // 조원: 김대현,
18 김재민, 박준홍;
19
20
21



‘주제 선정 이유’

◀ Programming Plan

- 01_ 주제 선정 이유
- 02_ API
- 03_ API 결과
- 04_ 협업 툴, 제작 도구
- 05_ 프롬프트
- 06_ 사이트 소개
- 07_ 마무리

직업별 임금



“이 직업은 얼마를 벌까?”
라는 단순한 생각에서 출발

API

Programming Plan

- 01_ 주제 선정 이유
- 02_ API
- 03_ API 결과
- 04_ 협업 툴, 제작 도구
- 05_ 프롬프트
- 06_ 사이트 소개
- 07_ 마무리

워크넷 직업정보 API

API 선정

- 공공데이터포털 API -> 데이터가 적음.
- 워크넷 직업정보 API -> 데이터 충분.

API 내용

- 하는 일, 임금, 직업 만족도, 전망, 성격, 가치관 등
- 직업에 관한 상세한 내용까지 있음.

API 결과

Programming Plan

- 01_ 주제 선정 이유
- 02_ API
- 03_ API 결과
- 04_ 협업 툴, 제작 도구
- 05_ 프롬프트
- 06_ 사이트 소개
- 07_ 마무리

```
<jobsList>
  <total>537</total>
  <jobList>
    <jobClcd>011</jobClcd>
    <jobClcdNM>의회의원 · 고위공무원 및 기업 고위임원</jobClcdNM>
    <jobCd>K000000847</jobCd>
    <jobNm>기업고위임원</jobNm>
  </jobList>
  <jobList>
    <jobClcd>011</jobClcd>
    <jobClcdNM>의회의원 · 고위공무원 및 기업 고위임원</jobClcdNM>
    <jobCd>K000000933</jobCd>
    <jobNm>행정부고위공무원</jobNm>
  </jobList>
  <jobList>
    <jobClcd>012</jobClcd>
    <jobClcdNM>행정 · 경영 · 금융 · 보험 관리자 </jobClcdNM>
    <jobCd>K000000919</jobCd>
    <jobNm>마케팅 · 광고 · 홍보관리자</jobNm>
  </jobList>
  <jobList>
    <jobClcd>012</jobClcd>
    <jobClcdNM>행정 · 경영 · 금융 · 보험 관리자 </jobClcdNM>
    <jobCd>K000001059</jobCd>
    <jobNm>경영지원관리자</jobNm>
  </jobList>
```

```
<salProspect>
  <jobCd>K000007525</jobCd>
  <jobLrcNm>경영 · 사무 · 금융 · 보험직</jobLrcNm>
  <jobMdcNm>회계 · 세무 · 감정 전문가</jobMdcNm>
  <jobSmcNm>세무사</jobSmcNm>
  <sal>조사년도: 2023년, 임금 하위(25%) 5775만원, 평균(50%) 8500만원, 상위(25%) 10000만원</sal>
  <jobSatis>80.6</jobSatis>
  <jobProspect>향후 5년간 세무사의 고용은 다소 증가하는 수준이 될 것으로 전망된다. 「2015-2025 한 세무사는 11,576명으로 2014년에 비해 각각 1천여 명 증가하였다. 통계청의 「전국사업체조사」에 만 아니라 일반인들도 소득에 대한 세금, 부동산 취득 및 양도 등을 비롯해 세금에 많은 관심을 갖고 늘어나고 있어, 전문 세무서비스에 대한 수요가 증가할 전망이다. 전문세무서비스에 대한 기업의 이용할 전망이다. 한편으로는 매년 일정 규모의 최소인원이 배출됨으로써 수입거래처가 획기적으로 늘 무량도 크게 늘어나고 있어, 장기적으로는 세무사의 고용감소를 가져올 가능성도 있으나 미는 전문
  <jobSumProspect>
    <jobProspectNm>증가</jobProspectNm>
    <jobProspectRatio>0</jobProspectRatio>
    <jobProspectInqYr>2023</jobProspectInqYr>
  </jobSumProspect>
```

· 협업 툴, 제작 도구

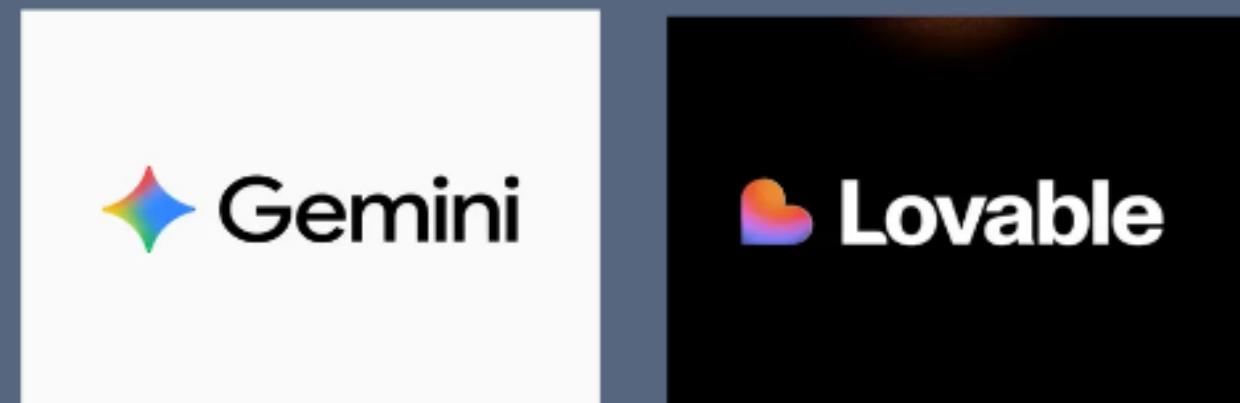
◀ Programming Plan

- 01_ 주제 선정 이유
- 02_ API
- 03_ API 결과
- 04_ 협업 툴, 제작 도구
- 05_ 프롬프트
- 06_ 사이트 소개
- 07_ 마무리

협업 툴



제작 도구



프롬프트

Programming Plan

01_ 주제 선정 이유

02_ API

03_ API 결과

04_ 협업 툴, 제작 도구

05_ 프롬프트

06_ 사이트 소개

07_ 마무리

1.

- React + Tailwind로 공공데이터 API 시각화 웹사이트 제작
- “데이터 불러오기” 버튼 → API 호출 → JSON 파싱 → 화면 출력
- 로딩 스피너 / 에러 메시지 / 카드·리스트 렌더링 구현
- API URL·Service Key 사용, CORS 시 프록시 등 우회 적용

2.

- React + Tailwind로 직업 리스트 → 임금 상세 조회까지 연동한 직업별 임금 대시보드를 구현한다.
- XML API에서 하위·평균·상위 임금을 불러와 카드 형태로 시각화하고 로딩·에러·CORS 처리를 적용한다.

3.

- 직업 코드 리스트와 선택한 jobCd의 상세 임금(XML)을 모두 <https://api.allorigins.win/raw?url=> 프록시로 호출해 파싱한다.
- 프록시 실패 시에는 임시(Mock) 데이터로 대체해 하위·평균·상위(25%)를 카드로 시연하며, 로딩·에러 처리를 포함한다.

프롬프트

Programming Plan

- 01_ 주제 선정 이유
- 02_ API
- 03_ API 결과
- 04_ 협업 툴, 제작 도구
- 05_ 프롬프트
- 06_ 사이트 소개
- 07_ 마무리

4.

- XML에서 <sal> 태그를 추출해 숫자만 남기고 정수로 변환한 뒤 오름차순 정렬한다.
- 정렬된 값으로 하위(첫 값)·평균(중간)·상위(마지막) 임금을 카드에 표시한다.
- 데이터가 부족하면 보정값 또는 Mock 데이터를 사용해 화면 출력이 항상 가능하도록 처리한다.

5.

- XML의 <sal> 값을 문자열로 불이지 말고, 숫자로 변환해 배열 wages[]에 push한다.
- 모든 값을 배열에 담은 뒤 오름차순 정렬하여 하위·평균·상위 임금을 인덱스로 가져온다.
- 핵심은 문자열 결합(+=)을 제거하고 배열 기반 파싱으로 구조를 단순화하는 것이다.

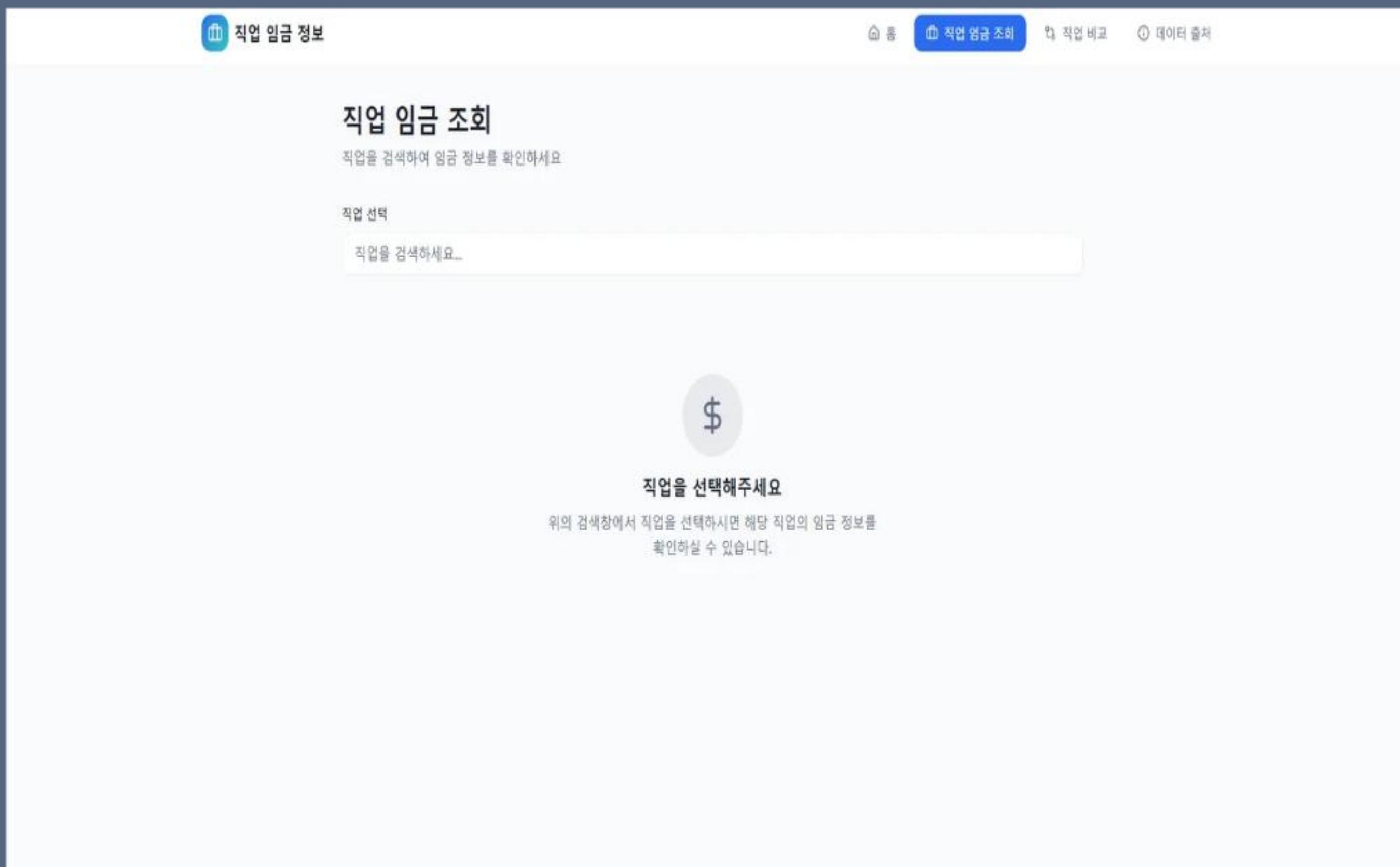
6.

- <sal> 태그 안의 문자열에서 정규표현식(Regex)으로 하위·평균·상위 임금 숫자를 각각 추출한다.
- 각 패턴(하위/평균/상위) 뒤의 숫자를 찾아 쉼표 제거 후 정수로 변환해 카드에 매핑한다.
- 패턴을 찾지 못하면 0 또는 “정보 없음”으로 처리해 화면 출력이 항상 가능하도록 만든다.

사이트 소개

Programming Plan

- 01_ 주제 선정 이유
- 02_ API
- 03_ API 결과
- 04_ 협업 툴, 제작 도구
- 05_ 프롬프트
- 06_ 사이트 소개
- 07_ 마무리



마무리

Programming Plan

- 01_ 주제 선정 이유
- 02_ API
- 03_ API 결과
- 04_ 협업 툴, 제작 도구
- 05_ 프롬프트
- 06_ 사이트 소개
- 07_ 마무리

소감

박준홍: api 연동을 처음하면서 우여곡절도 많았지만 다같이 머리를 풀어나가면서 재미있다 란 생각이 들어서 좋았습니다. "API 연동을 처음 하면서 우여곡절도 많았지만, 다 같이 머리를 맞대고 문제를 풀어나가면서 재미있다는 생각이 들어서 좋았습니다."

김대현: AI를 활용하여 웹사이트를 만드는 작업과 팀을 이루어 협업을 통해 프로젝트를 수행하는 과정에서 커뮤니케이션의 중요성을 깨닫게 되었습니다.

김재민: 프론트엔드 영역에서 데이터를 가공하고 이를 최종 서비스 형태로 구현하는 과정을 경험했습니다., 프론트엔드에서 데이터 가공부터 서비스 완성에 이르는 전 과정을 경험할 수 있었습니다.

reference

https://i.namu.wiki/i/00YiJ77lufWZpb_7JrhCNAjI21fC-mknRsDsJiV1plpI0I65UhQKsqY7j0vk54Pljo4wMZxQoFXYFFF1STy4KSGWeZwt0WA3wLXaCkPpIvU4jmKYBgEvjyFHL9ImkvEH56PoXb_E6MLxmV8iHleyVQ.svg

https://i.namu.wiki/i/V2Tce7Xn7yrcQHf0KtjmC05hwAMc7IERptsv3C_4QjYDXrfva-Wr4hiEIyDOWrrDk2yjVR3KNGdUygtmYc1qVQ.svg

<https://designcompass.org/wp-content/uploads/2025/07/google-gemini-07-1536x1152.jpg>

<https://designcompass.org/wp-content/uploads/2025/04/Lovable2-10-1536x1152.jpg>