



Algoniion;

모든 코드, 모두 모아

(숨고르기) 안녕하세요.

**모든 코드, 모두 모아, 알고니온 서비스
의 발표를 맡은 000 입니다.**

**발표에 시작하기에 앞서,
노래를 한 곡 들려드리려고 합니다.**

(아이브-키치)

**(웃음) 지금부터 알고니온 서비스의
소개를 시작하겠습니다.**

온라인 저지 사이트

BAEKJOON
ONLINE JUDGE



CODE
TREE



programmers

/ Softeer.

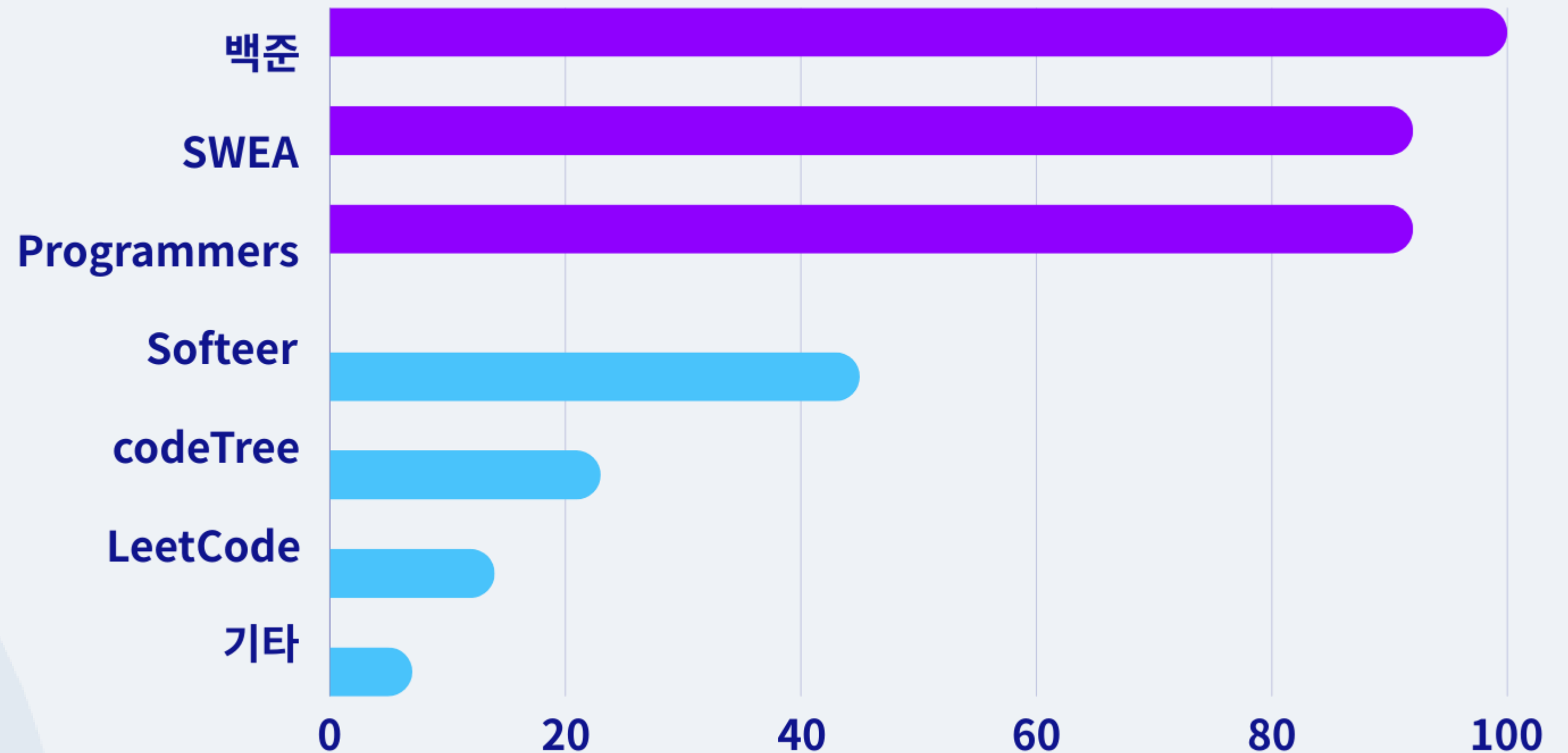
Code Up

**다들 온라인 저지 사이트를 사용해보셨죠?
(조금 숨돌리기)**

낯익은 이름이 많이 보이네요.

**각 사이트를 얼마나 많은 비중으로
이용하는지 궁금하지 않으신가요?**

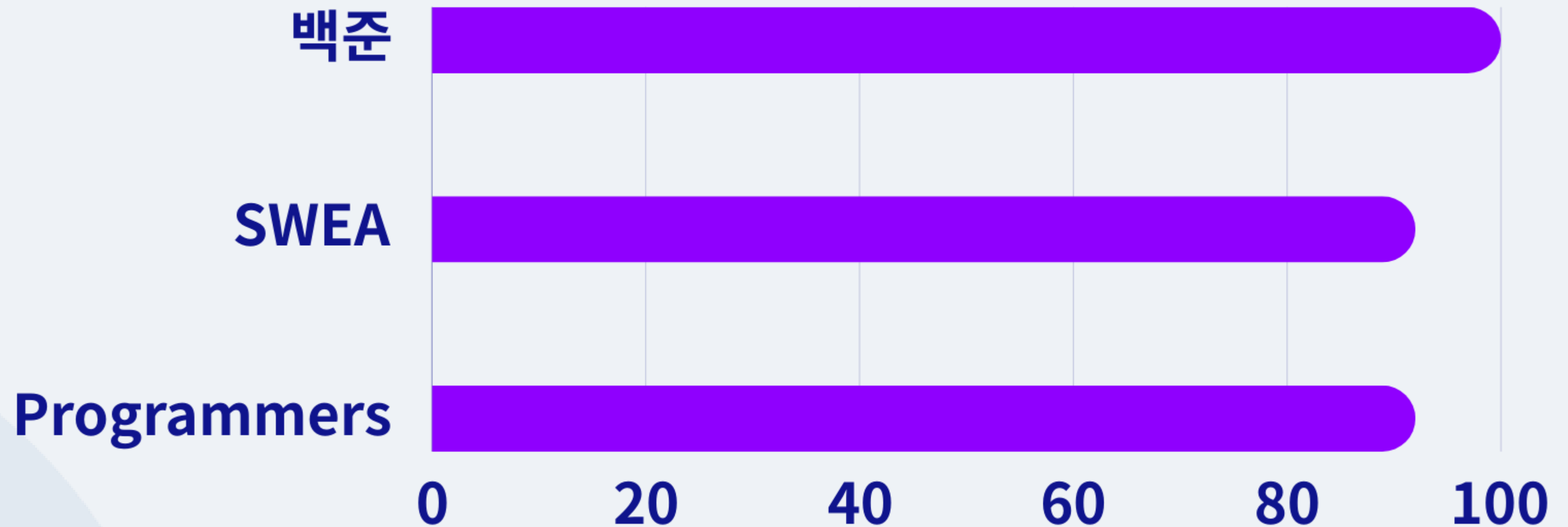
온라인 저지 사이트 사용 경험



저희는 온라인 저지 사이트의
사용 경험을 확인하기 위해,
싸피인들의 설문조사를 실시했었습니다.

보시다시피 다양한 사이트에서의
경험을 가지고 있고,

온라인 저지 사이트 사용 경험



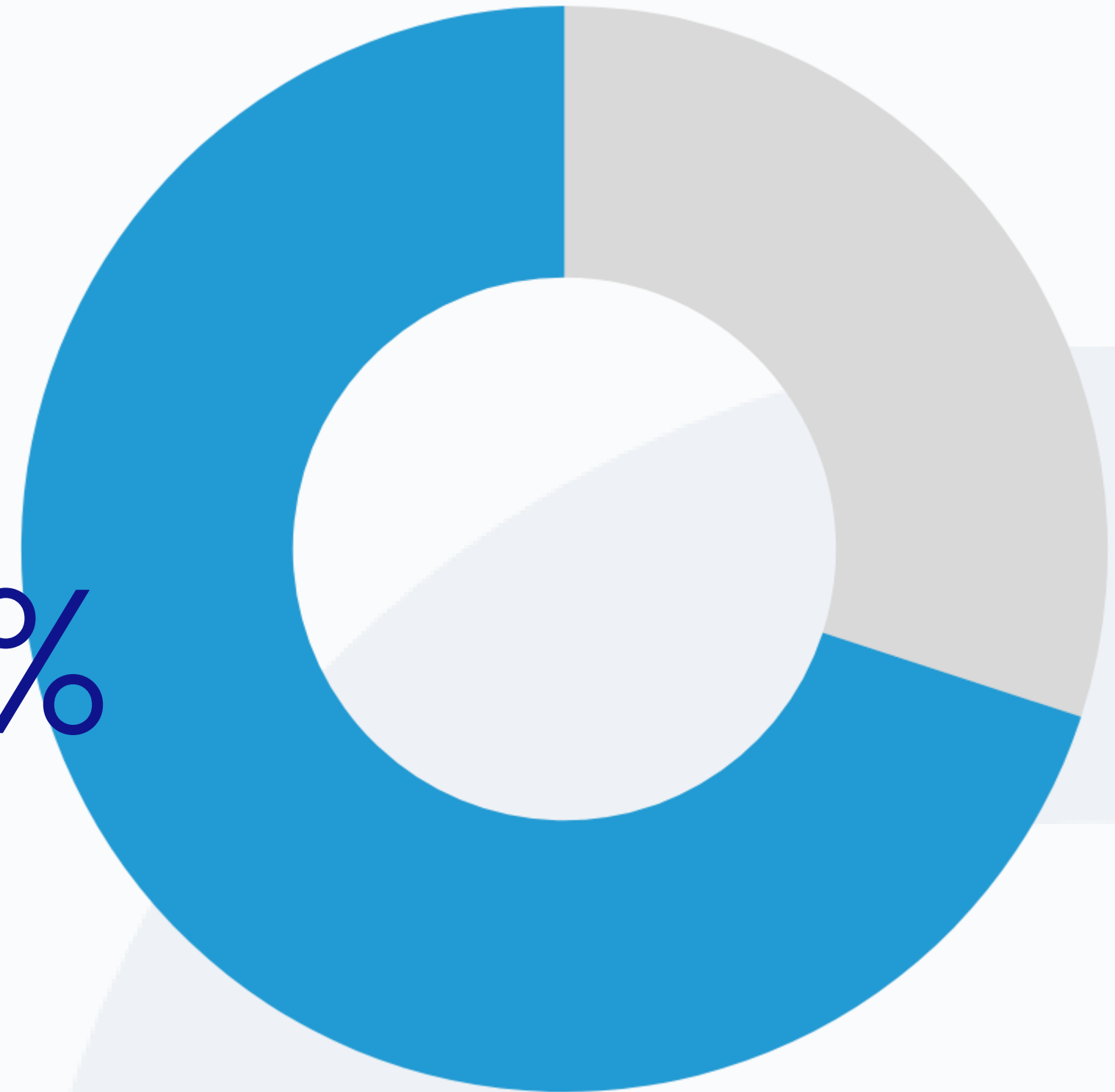
90% 이상

특히 백준/swea/프로그래머스의 사용
경험이 높았으며, 그 중,



2개 이상 사용 비율

70%



**2개 이상의 사이트를 이용하는
사피인들이 많았습니다.**

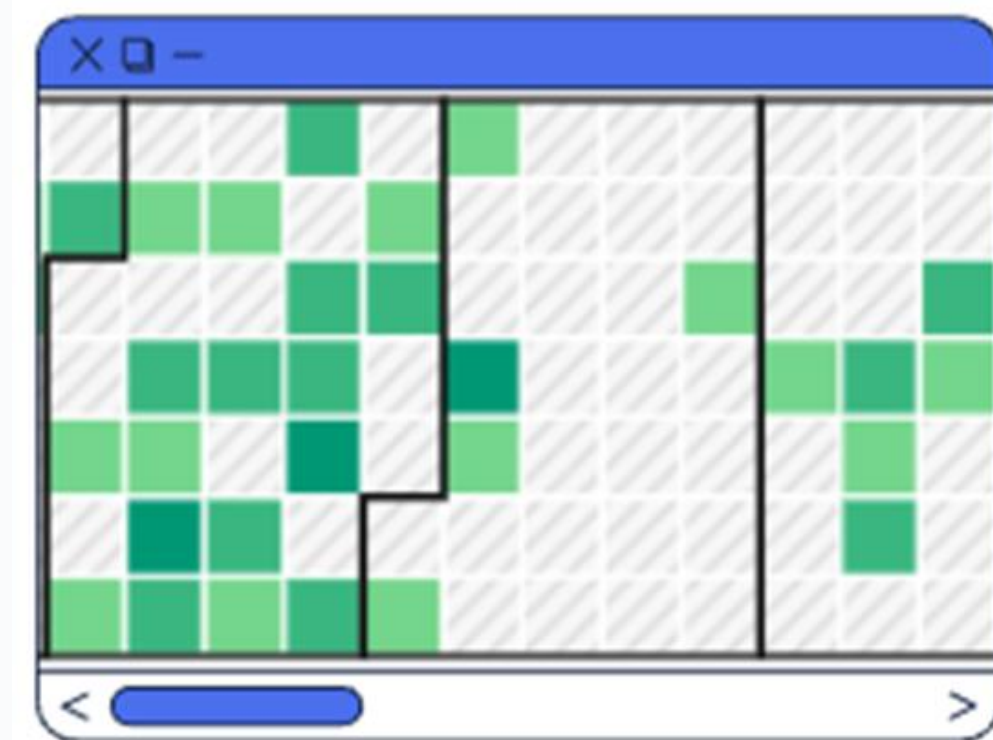
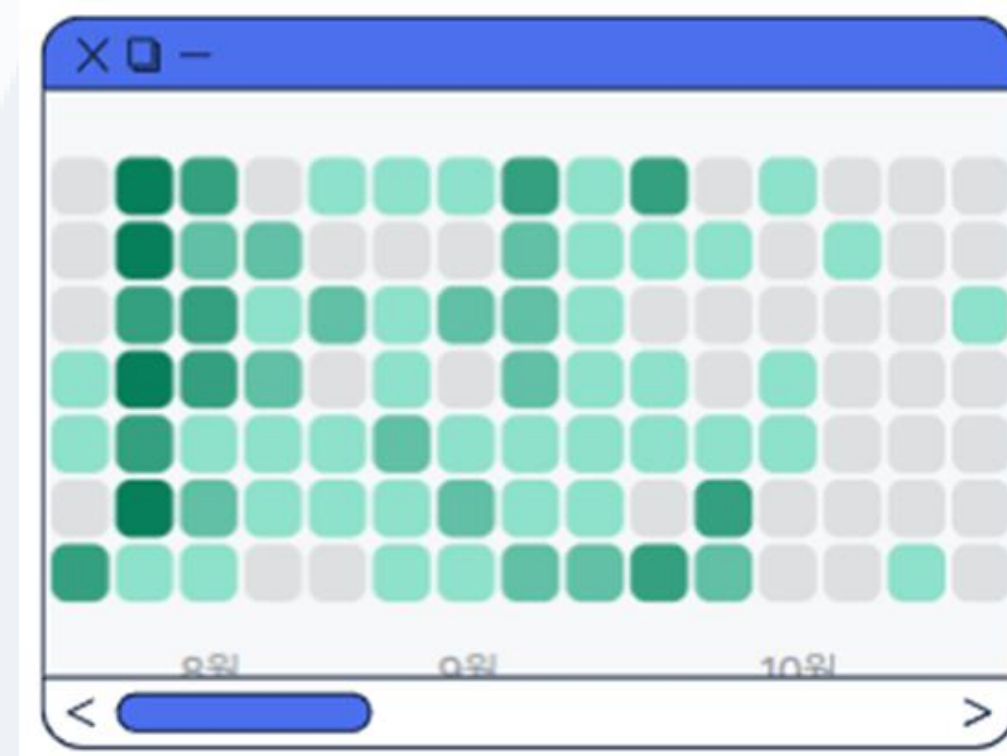
**또한 온라인 저지 사이트를
다중으로 이용하는 것에 대한
불편함이 있지 않을까 생각하여,**

추가적인 설문을 진행했습니다.

스트릭 공유

되지 않음

54%



각 사이트에서 스트릭이
공유되지 않는다는 점이
절반 이상으로,
가장 큰 불편함으로 꼽혔고,
그 다음으로는

실시간 피드백

불가

40%



문제를 풀면서 겪는

어려움에 대해

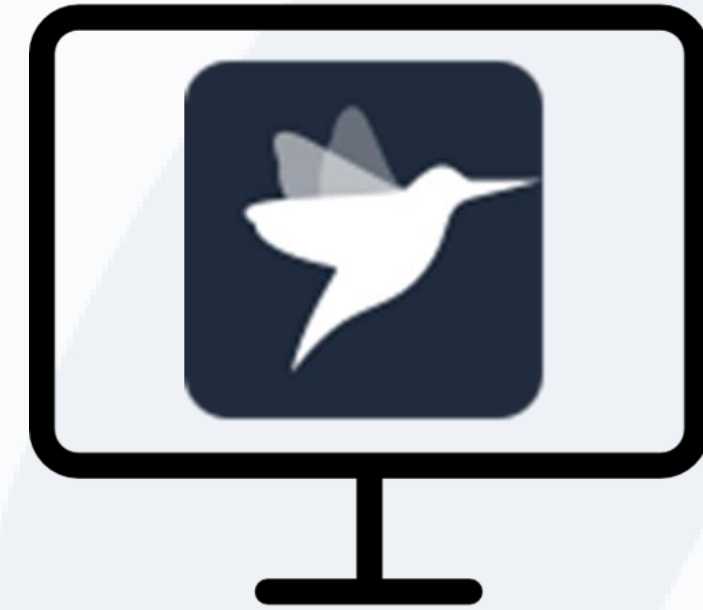
실시간으로

피드백을 받을 수 없다는 점 40%,

풀었던 문제

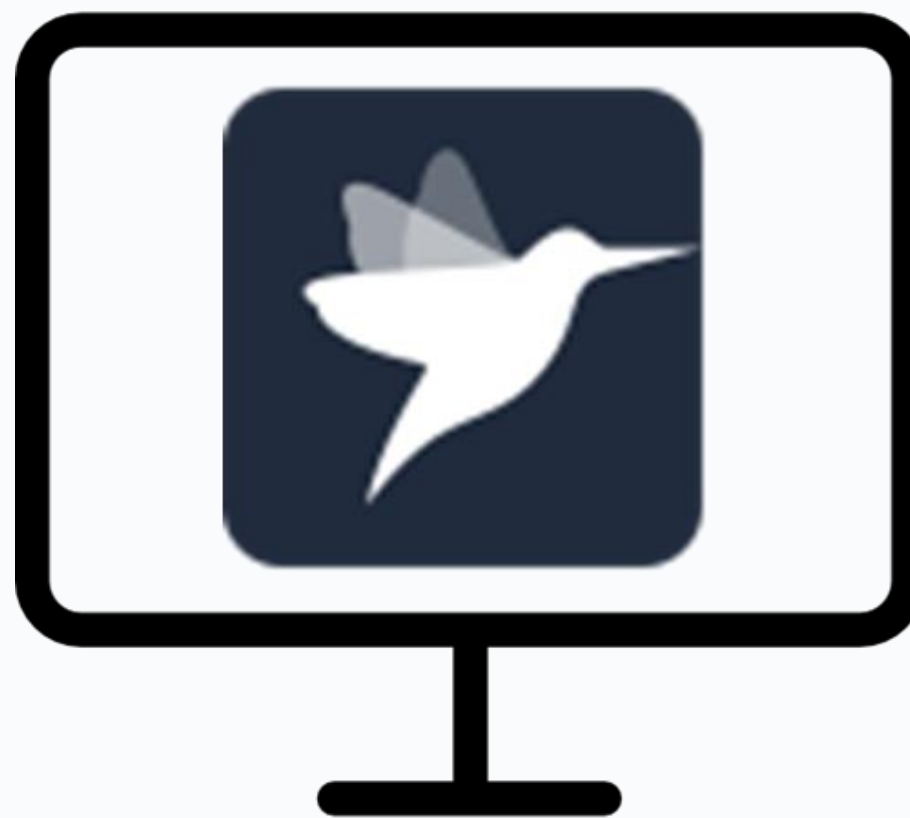
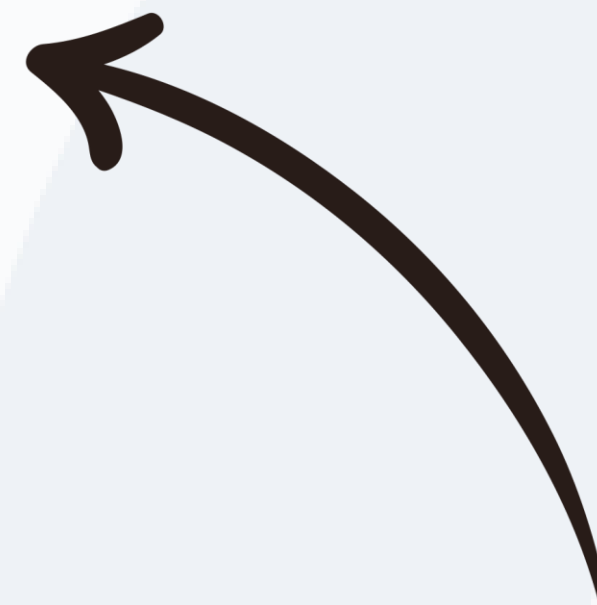
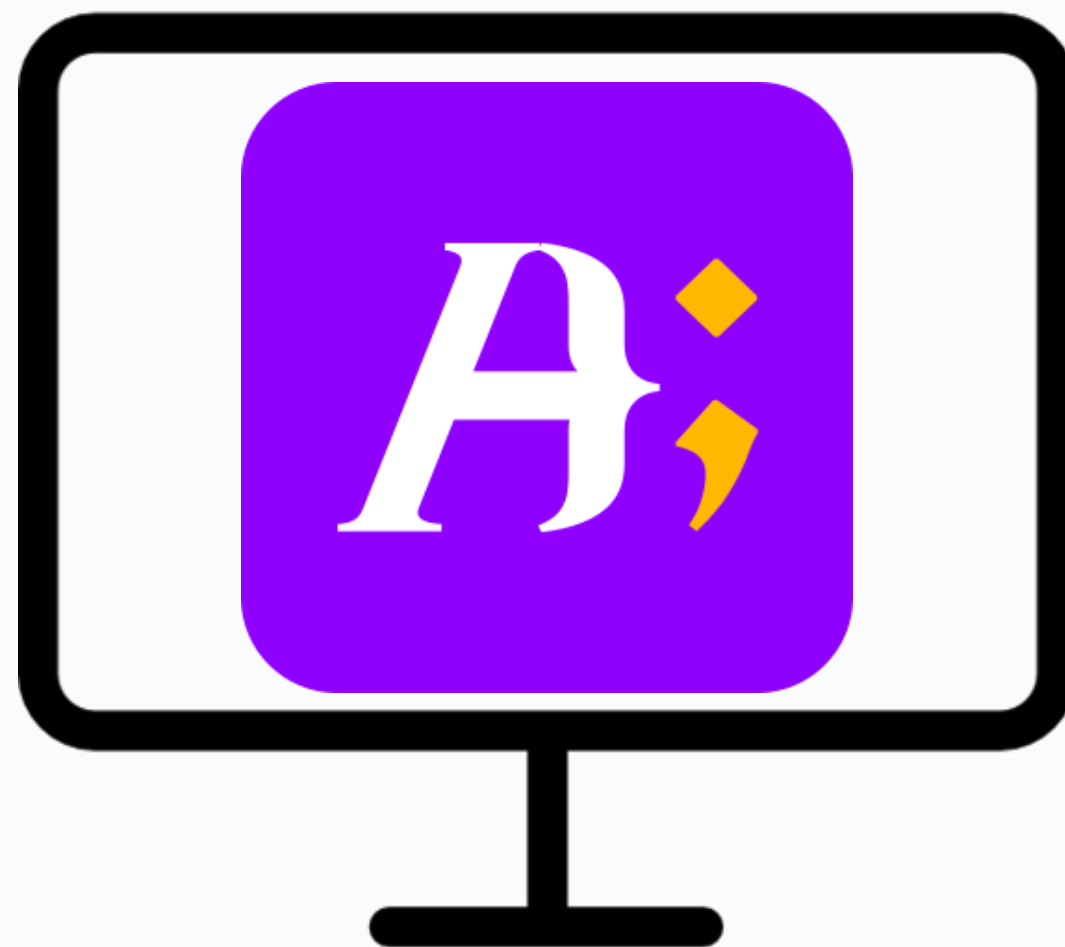
분산

38%



**풀었던 문제가 각 사이트에
분산되어있다는 점이,
주요 불편함으로 집계되었습니다.**

이러한 불편한 점을 해결하기 위해,



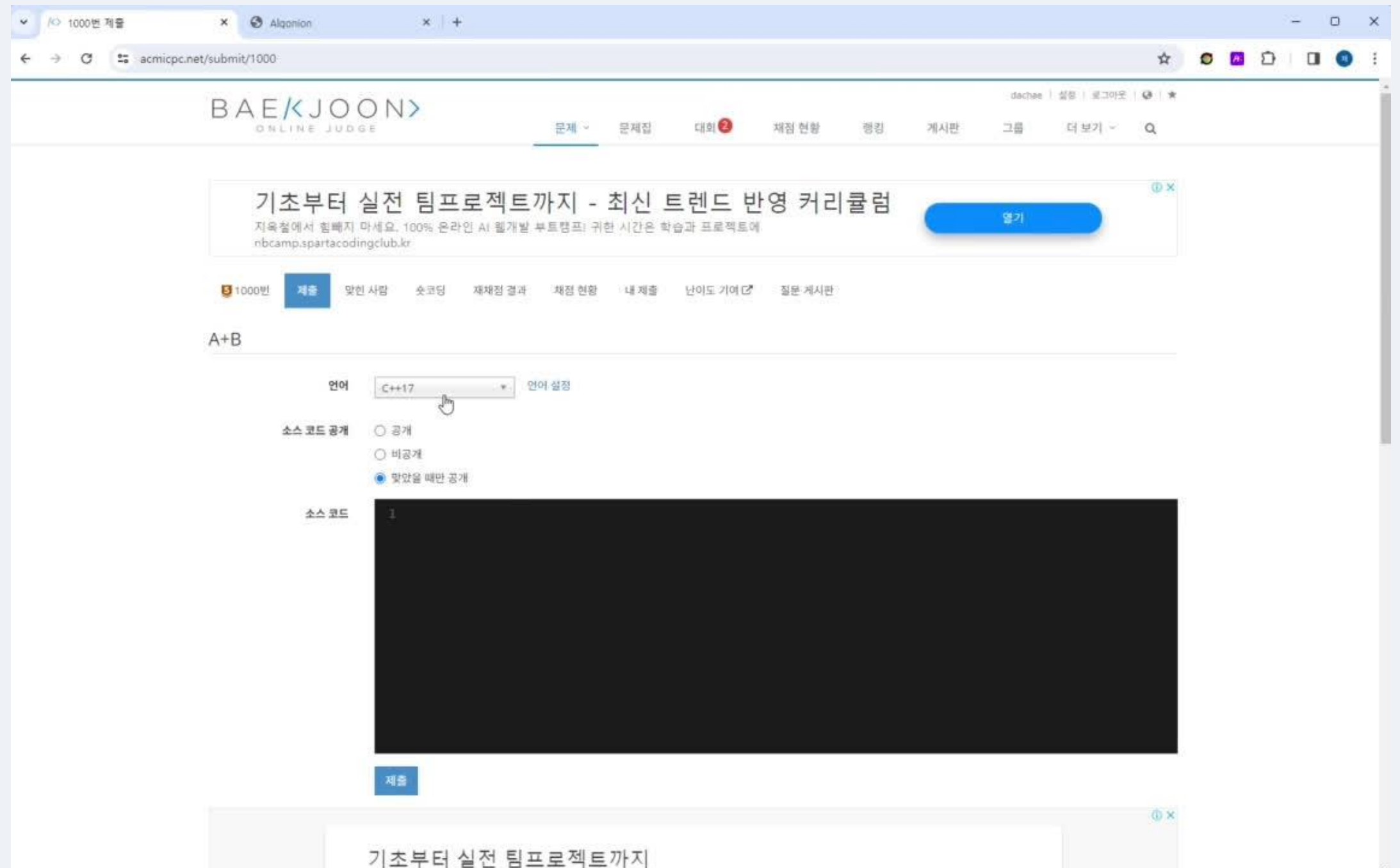
**이용률이 가장 높은 세 가지 사이트,
백준, 프로그래머스, swea 의 통합을
기획하게 되었습니다.**

백준

1. 최신 문제 채점 여부 확인

2. 채점 완료 시 정답 정보 파싱

3. DB에서 제출번호 중복확인



우선, 알고니온 서비스에서는
각 문제풀이에 대한
데이터를 가져옵니다.

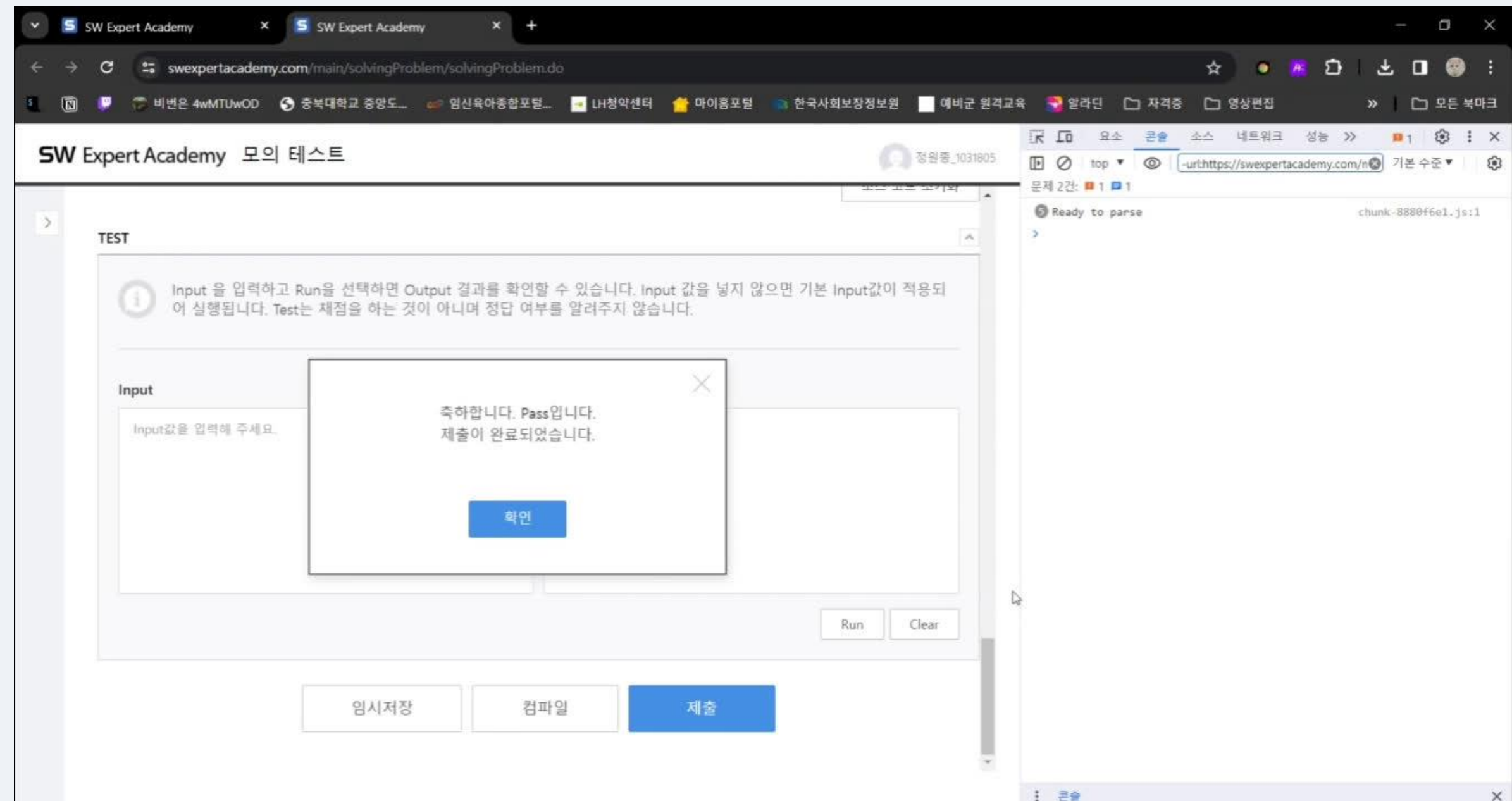
백준의 경우에는
'내 제출 페이지' 에서 현재 최신 문제가
채점되고 있는지 확인하고, 기다립니다.

**채점이 모두 끝났다면,
정답 정보 중에
가장 최신 데이터를 파싱하고,
DB에서 제출 번호가 중복인지 확인하여
저장하게 됩니다.**

Programmers & SWEA

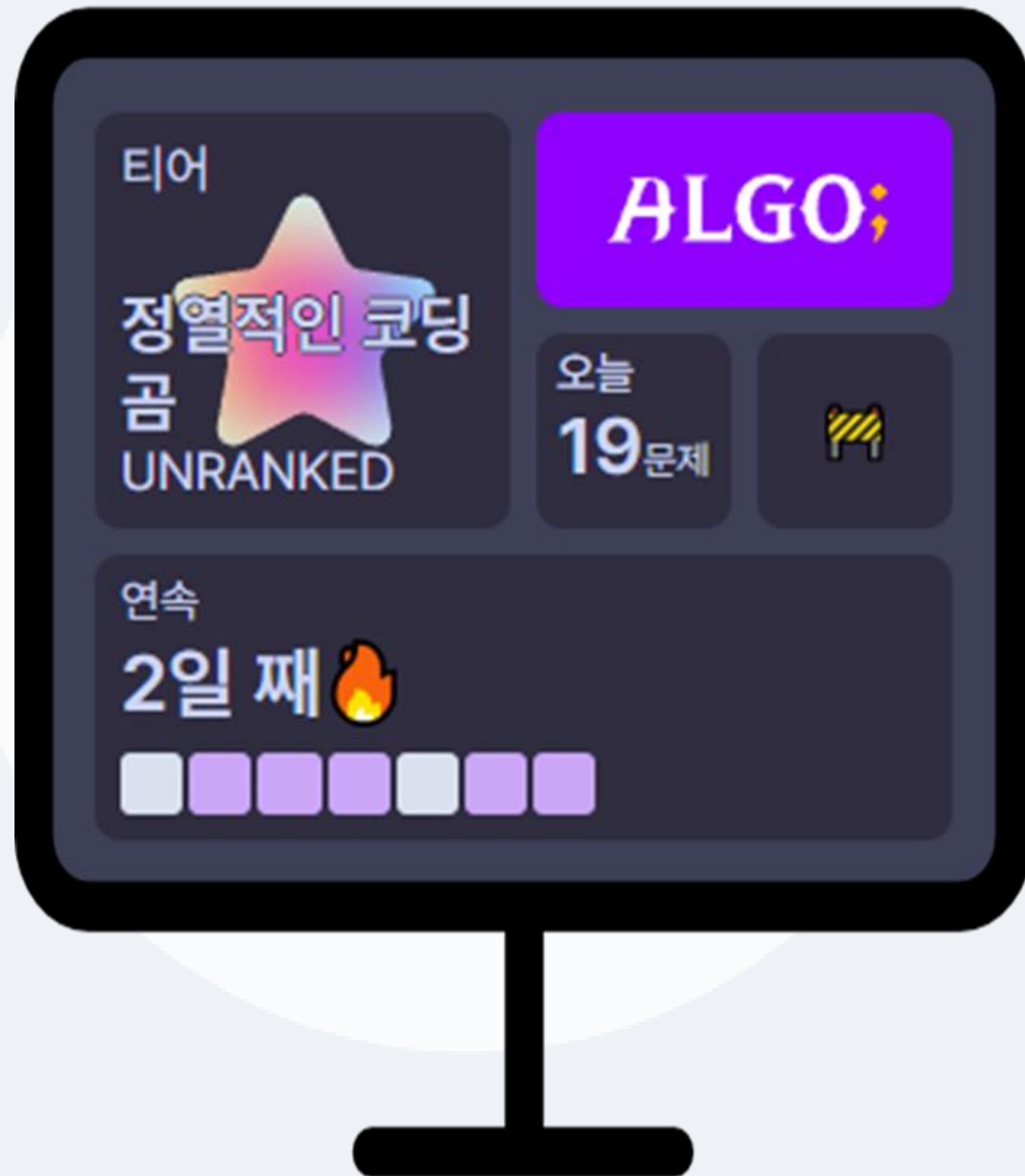
1. 채점 여부 확인

2. 채점 완료 시 정답 정보 파싱



**프로그래머스와 SWEA 의 경우에는
'문제 페이지' 에서 풀이를 제출하고,
정답이면
해당 데이터를 DB에 저장합니다.**

**이러한 일련의 과정들은 크롬 익스텐션
을 통해 진행됩니다**



확장 프로그램

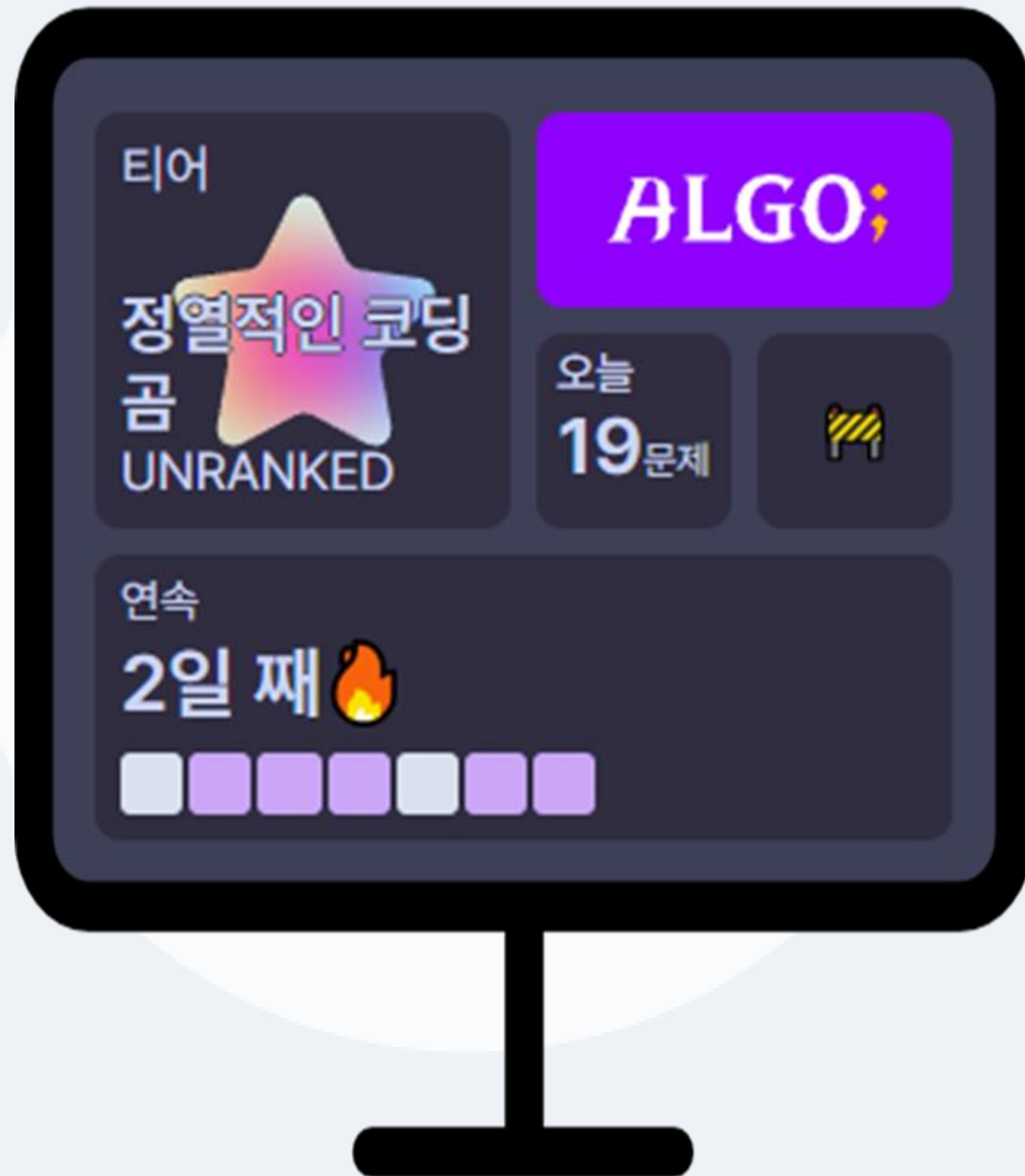
Mutation

Observer

**크롬 익스텐션에서 사용한
mutation observer 기술은
브라우저의 DOM 변경사항을
감지합니다.**

**현재 백준허브에서
백준에 사용하는 파싱 방식은**

**2초에 한 번씩 브라우저의 변화를
확인하는 것에 비해 더 진화한 방식
(or 똑똑한 방식) 으로 볼 수 있습니다.**



확장 프로그램

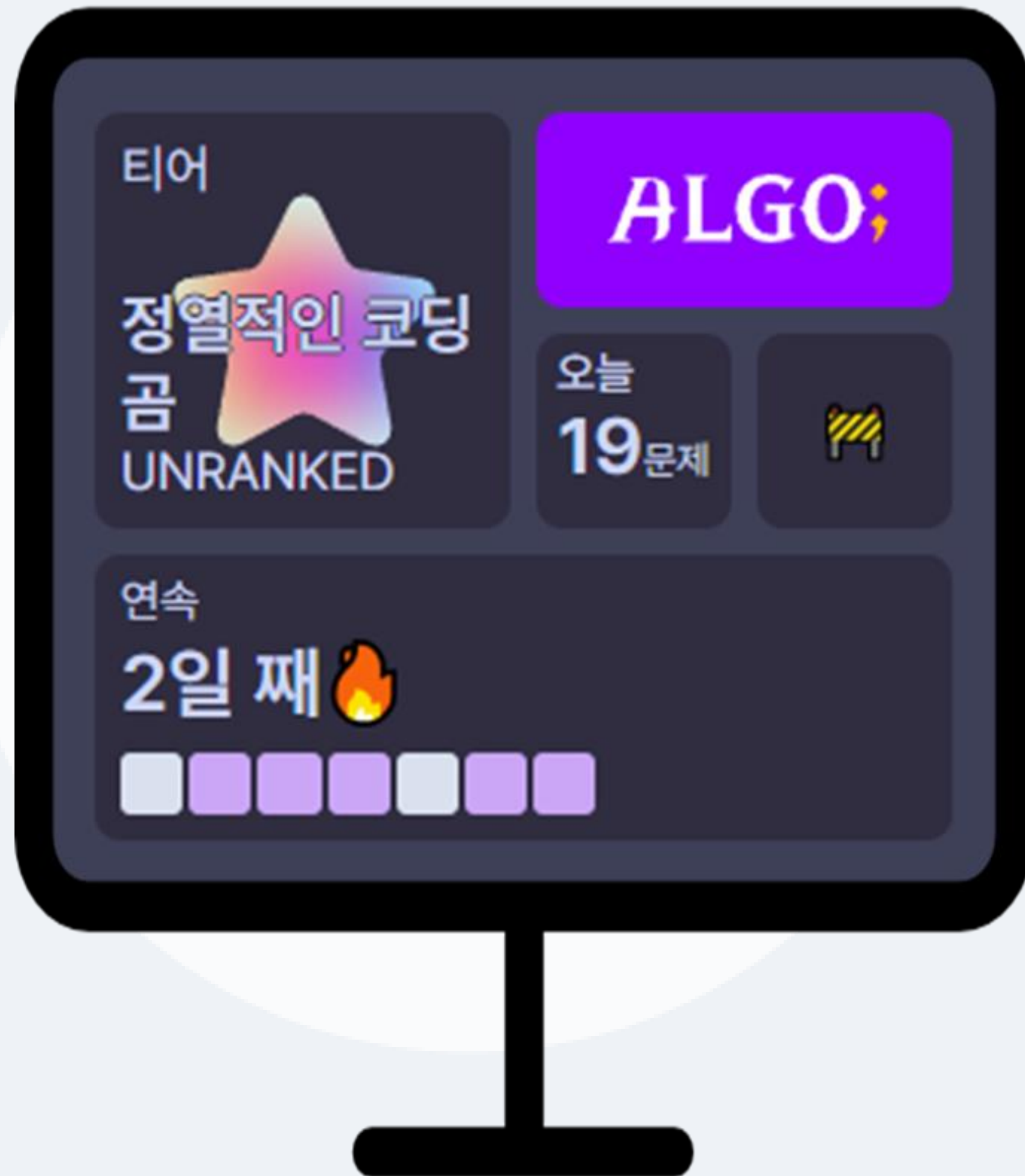
편리함, 시간 절약

정확성, 통일성

**크롬 익스텐션을 사용하는 게,
더 귀찮은 것이 아닌가 생각하는 분들이
계실 수도 있을텐데요.**

편리함

**첫 째, 크롬 익스텐션을 사용하면,
사용자가 따로 정보를 입력할 필요 없이,
자동으로 데이터를 저장할 수 있습니다.**

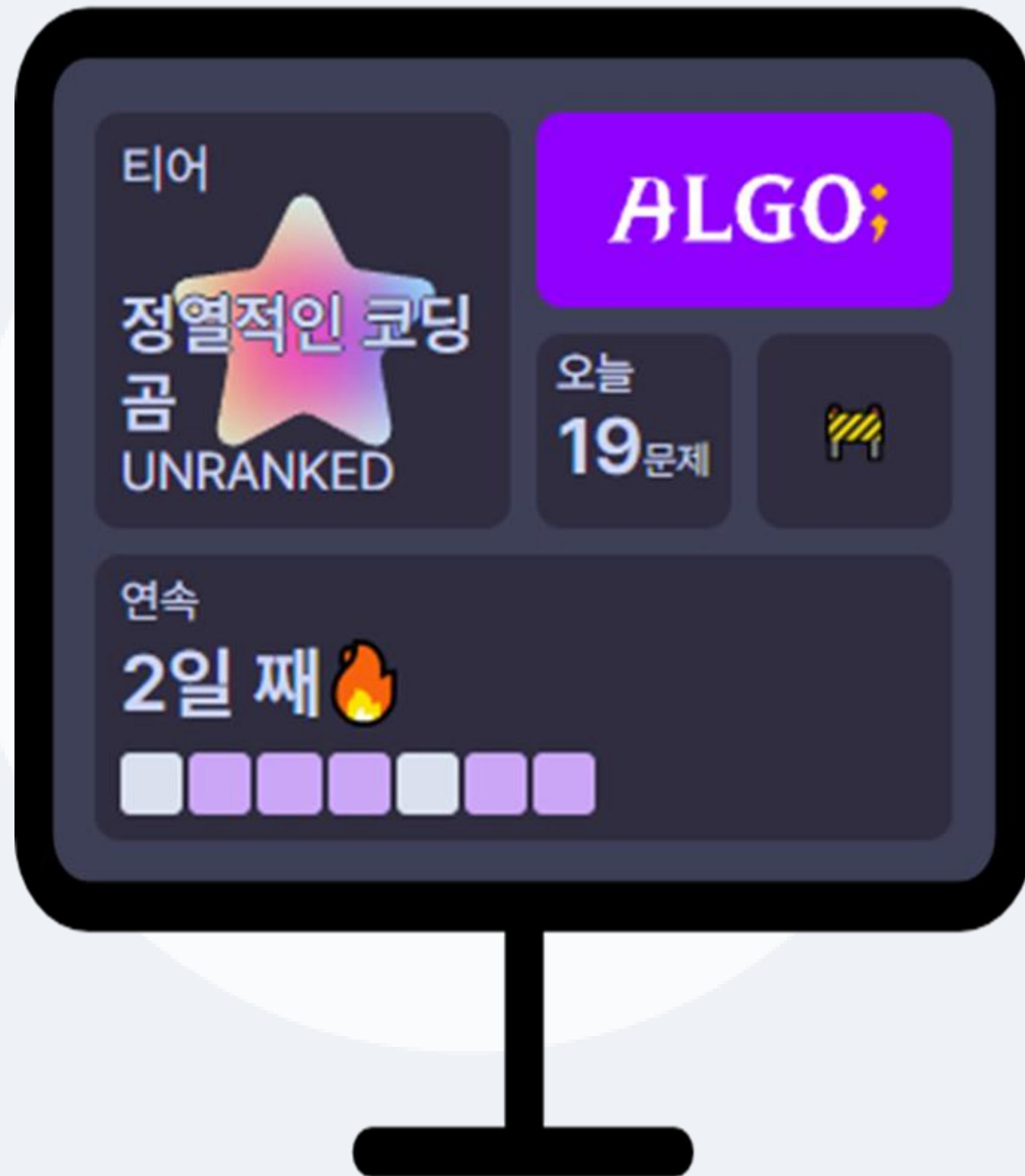


확장 프로그램

편리함, 시간 절약

정확성, 통일성

**둘째,
저장하고 싶은 풀이를 각 사이트에
접속하며
수동으로 찾아오는 번거로움을
줄여줍니다.**



확장 프로그램

편리함, 시간 절약

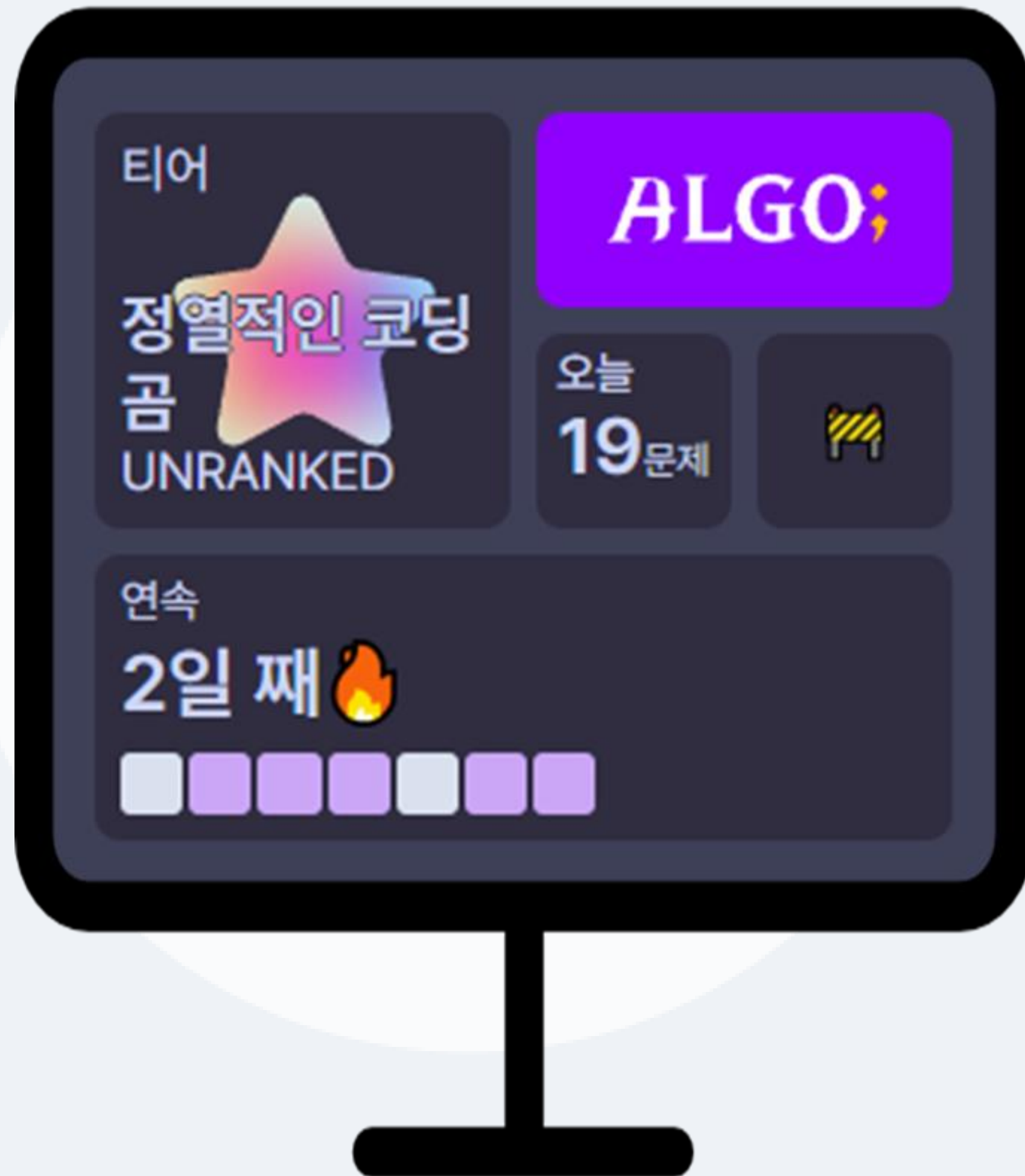
정확성, 통일성

셋째,

수기 입력을 통해 발생할 수 있는

데이터의 오류를 방지하여,

데이터의 정확성을 높여줍니다.



확장 프로그램

편리함, 시간 절약

정확성, 통일성

**마지막으로는
동일한 방식으로 정보를
가져와 저장하므로,
모든 데이터가 일관되고 통일성있게
저장됩니다.**

BAE<KJOON>



Algonion;

나의정보 코드로그 커뮤니티 Q 유저검색



ALGO 님의 코드로그

문제 추천

제 목	레 벨	언 어	제 출 일	비 고
2001. 파리 퇴치	D2 Swea	Python	2024-01-27	
1949. [모의 SW 역량테스트] 등산로 조성	Unrated Swea	JAVA	2024-01-27	
1592. 영식이와 친구들 구현, 시뮬레이션	Bronze II Baekjoon	Java 11	2024-02-03	
1697. 숨바꼭질 너비 우선 탐색, 그래프 이론, 그래프 탐색	Silver I Baekjoon	Java 11	2024-02-04	
1764. 들보잡 자료 구조, 해시를 사용한 집합과 맵, 정렬, 문자열	Silver IV Baekjoon	Python 3	2024-02-04	
1874. 스택 수열 자료 구조, 스택	Silver II Baekjoon	Python 3	2024-02-04	
1890. 점프 다이나믹 프로그래밍	Silver I Baekjoon	PyPy3	2024-02-05	

**크롬 익스텐션을 통해
각 사이트의 풀이는
알고니온의 코드로그 게시판에
한 눈에 볼 수 있도록 저장됩니다.**

**이는 설문조사에서 나왔던
각 사이트에서 푼 문제를 찾기 어렵다는
문제를 해결할 수 있습니다.**



ALGO 님의 코드로그

문제 추천

제 목	레 벨	언 어	제 출 일	비 고
2001. 파리 퇴치	D2 Swea	Python	2024-01-27	
1949. [모의 SW 역량테스트] 등산로 조성	Unrated Swea	JAVA	2024-01-27	
1592. 영식이와 친구들 구현, 시뮬레이션	Bronze II Baekjoon	Java 11	2024-02-03	
1697. 숨바꼭질 너비 우선 탐색, 그래프 이론, 그래프 탐색	Silver I Baekjoon	Java 11	2024-02-04	
1764. 들보잡 자료 구조, 해시를 사용한 집합과 맵, 정렬, 문자열	Silver IV Baekjoon	Python 3	2024-02-04	
1874. 스택 수열 자료 구조, 스택	Silver II Baekjoon	Python 3	2024-02-04	
1890. 점프 다이나믹 프로그래밍	Silver I Baekjoon	PyPy3	2024-02-05	

Swea
2001

파리 퇴치

```
T = int(input())

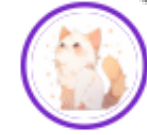
for test_case in range(1, T+1):
    N, M = map(int, input().split())
    board = [list(map(int, input().split())) for i in range(N)]
    maxBug = 0
    for rows in range(N):
        if(rows + M > N):
            break
        for column in range(N):
            temp = 0
            if(column + M > N):
                break
            for row in range(rows, rows + M):
                temp += sum(board[row][column:column+M])
            if(maxBug < temp):
                maxBug = temp
    print('#{} {}'.format(test_case, maxBug))
```

Swea
2001

파리 퇴치

```
T = int(input())

for test_case in range(1, T+1):
    N, M = map(int, input().split())
    board = [list(map(int, input().split())) for i in range(N)]
    maxBug = 0
    for rows in range(N):
        if(rows + M > N):
            break
        for column in range(N):
            temp = 0
            if(column + M > N):
                break
            for row in range(rows, rows + M):
                temp += sum(board[row][column:column+M])
            if(maxBug < temp):
                maxBug = temp
    print('#{} {}'.format(test_case, maxBug))
```

```
T = int(input())

for test_case in range(1, T+1):
    N, M = map(int, input().split())
    board = [list(map(int, input().split())) for i in range(N)]
    maxBug = 0
    for rows in range(N):
        if(rows + M > N):
            break
        for column in range(N):
            temp = 0
            if(column + M > N):
                break
            for row in range(rows, rows + M):
                temp += sum(board[row][column:column+M])
            if(maxBug < temp):
                maxBug = temp
    print('#{0} {1}'.format(test_case, maxBug))
```



메모

- 문제 요약



```
for column in range(N):
    temp = 0
    if(column + M > N):
        break
    for row in range(rows, rows + M):
        temp += sum(board[row][column:column+M])
    if(maxBug < temp):
        maxBug = temp
print('#{0} {1}'.format(test_case, maxBug))
```

메모

- 문제 요약

- 사용한 알고리즘

·
·

-풀이



```
print('#{0} {1}'.format(test_case, maxBug))
```



메모

- 문제 요약

- 사용한 알고리즘

·
·

-풀이

SAVE

NOTION



메모

- 문제 요약

- 사용한 알고리즘

- 단순탐색

- 풀이

- 모든 행(rows)에 대해:
- M 크기만큼 범위가 배열 크기를 넘지 않는지 확인
 - 모든 열(column)에 대해:
 - 현재 위치에서 M 크기만큼 파리 퇴치 수 계산:
 - M 크기만큼 행을 반복하며 해당 열의 모든 값을 더해서 현재 퇴치 수(temp)에 누적
 - 최대 퇴치 수(maxBug)와 현재 퇴치 수(temp) 비교 및 갱신

노션 내보내기

ALGO

나의정보 코드로그 커뮤니티 Q 유저검색

메모

- 문제 요약

- 사용한 알고리즘

•

•

- 풀이

어려웠다...

24.02.15 17:20:28

SAVE

NOTION

Algonion

공유 ⌚ ☆ ⋮

이 페이지는 flying-wolfberry-583.notion.site에 게시되어 있습니다.

🌐 사이트 보기 ⚙️ 사이트 설정

Algonion

표 표 (1) 1개 더 보기

필터 정렬 ⚡ 🔍 ⋮

새로 만들기

Aa 문제 이름 + 필터 추가

초기화 모두에게 저장

📍 사이트 문제 번호 Aa 문제 이름 분류 URL

필터 결과가 없습니다. 행을 추가하려면 클릭하세요.

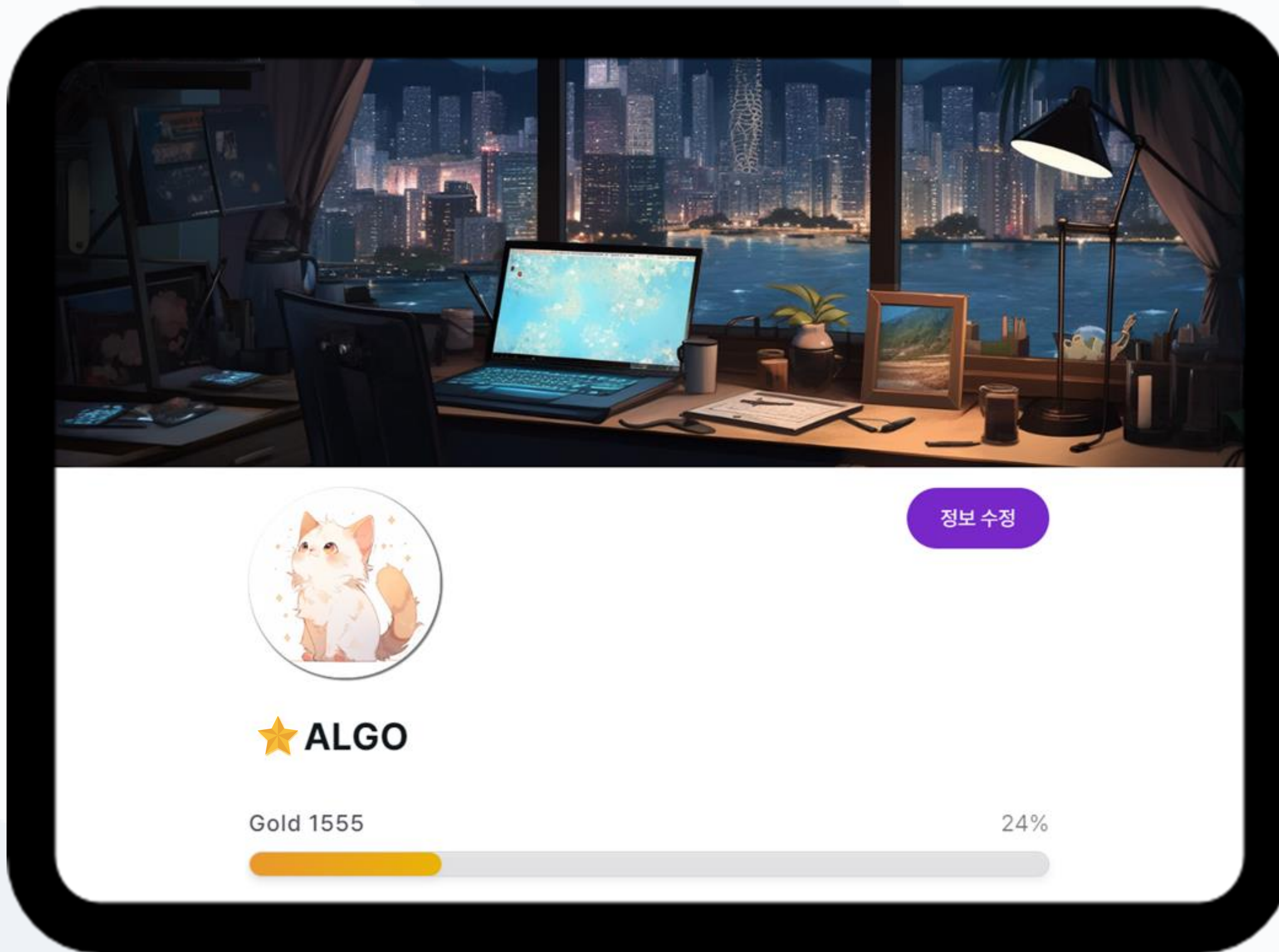
+ 새로 만들기

계산



서연

User 정보



A user profile card for a user named ALGO. The card features a large background image of a desk with a laptop and a view of a city at night. Below the image is a circular profile picture of a cat. To the right of the profile picture is a purple button labeled '정보 수정'. Below the profile picture is the username '★ ALGO'. At the bottom, there is a progress bar showing 'Gold 1555' and '24%'.

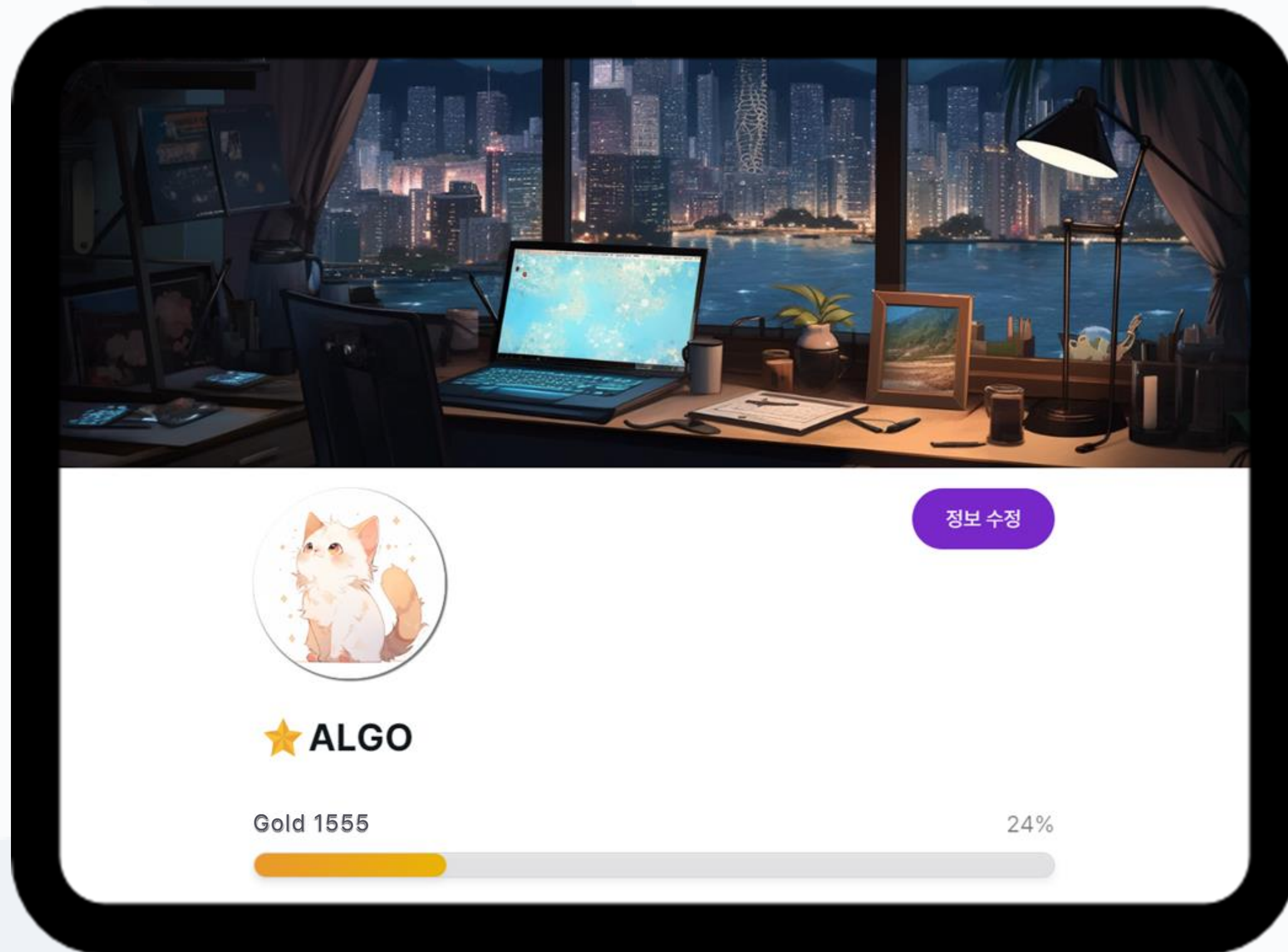
정보 수정

★ ALGO

Gold 1555 24%

User 정보

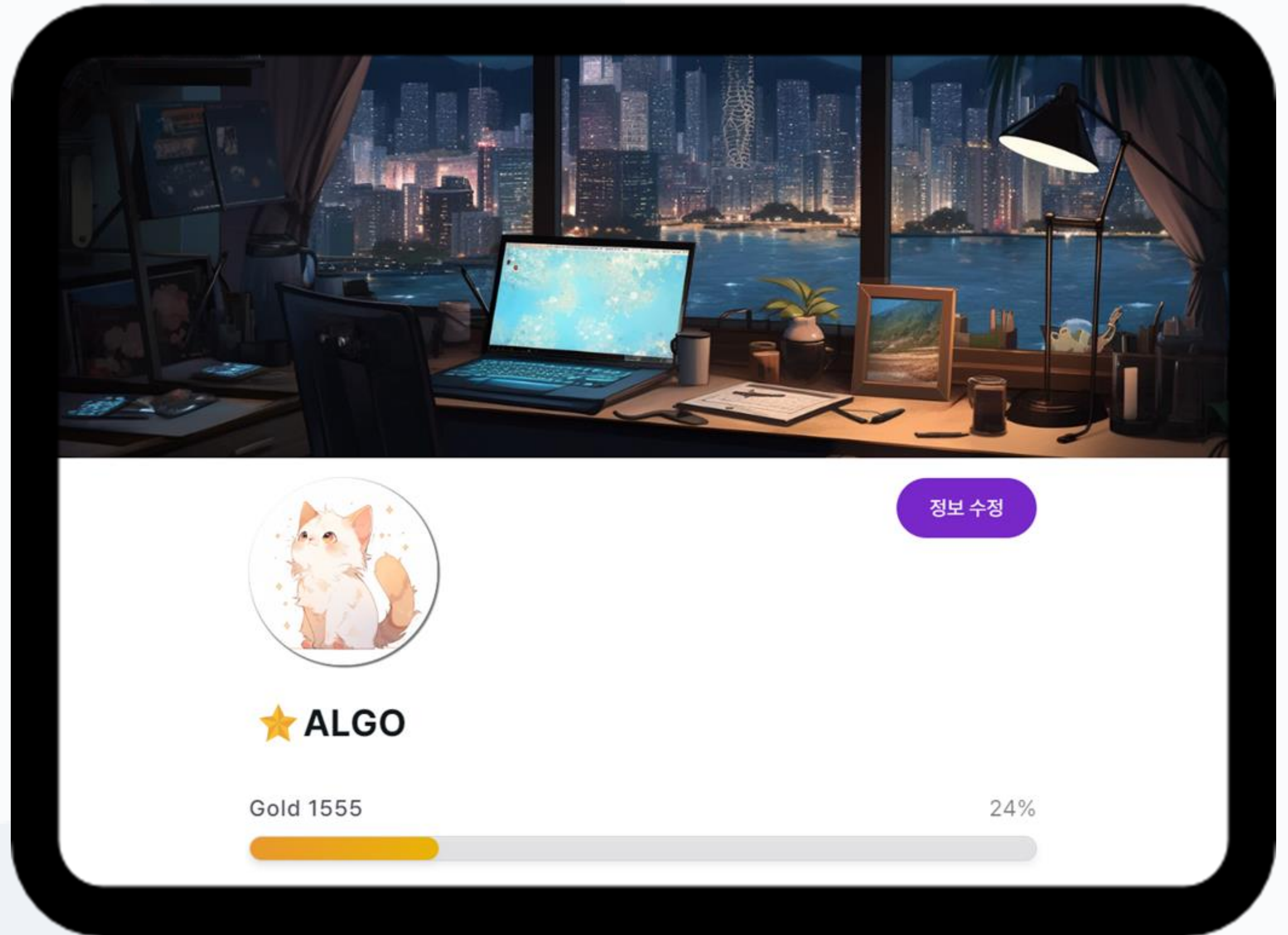
티어 제공



User 정보

통합 점수 제공

Gold 1555



TOP 100



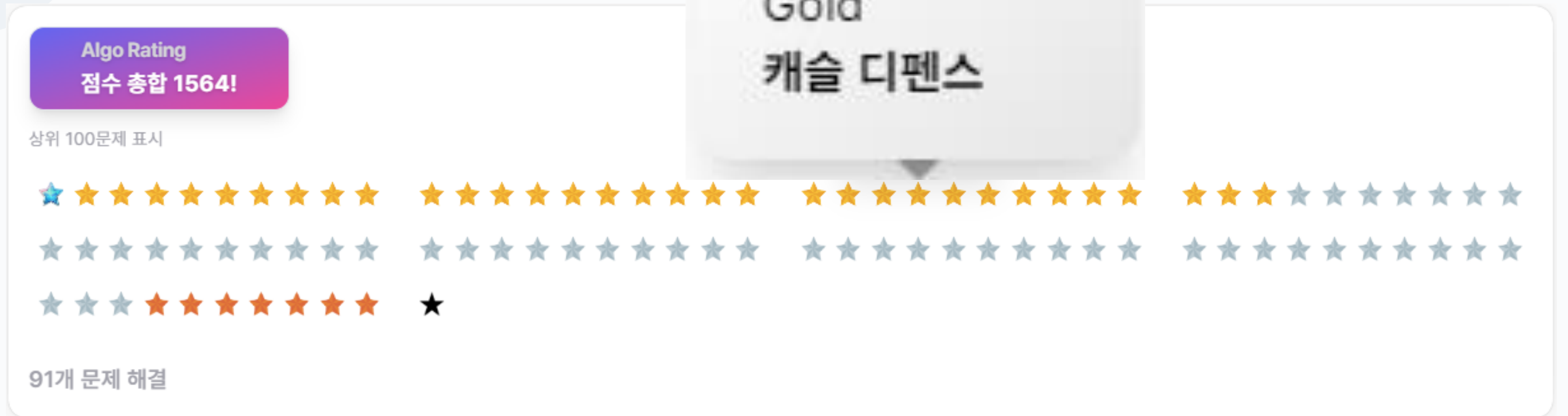
TOP_100 문제

호버_시_문제_정보

종합_점수

클릭_시_문제_이동

TOP 100



TOP_100 문제

호버_시_문제_정보

종합_점수

클릭_시_문제_이동

스트릭

Algo Streak
현재 3일!

1년 간의 스트릭이 표시됩니다.



TOP_100 문제

호버_시_문제_정보

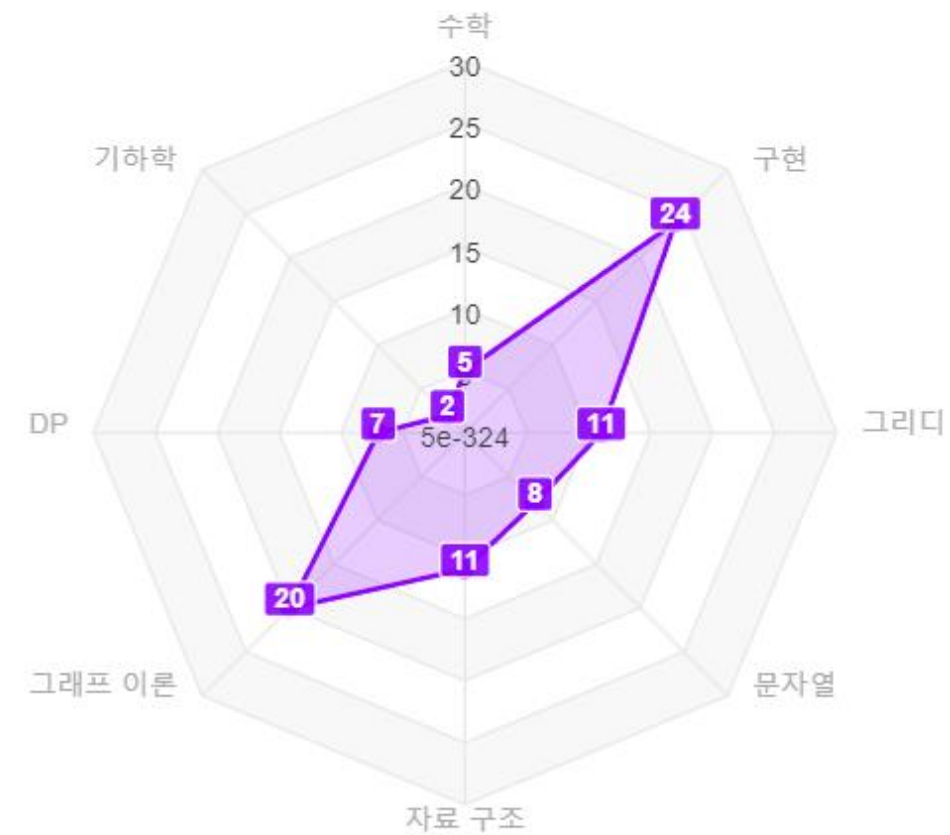
종합_점수

클릭_시_문제_이동

대시보드

- 문제 분류

기록의 여정, 육각형 개발자



문제_분류 # 방사형_그래프

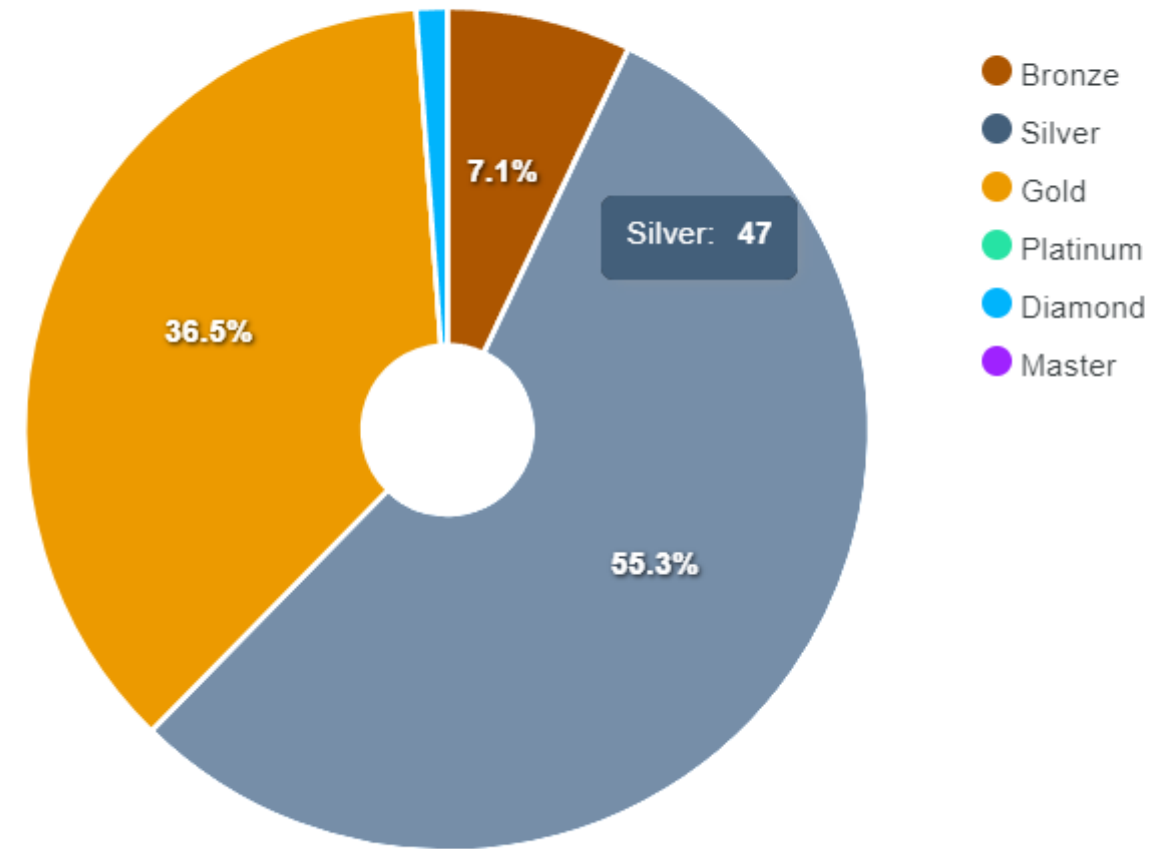
나_구현에_재능_있는_듯?

#기하학_공부가_필요해

대시보드

- 난이도 분류

기록의 여정, 문제 난이도



난이도_분류

원형_그래프

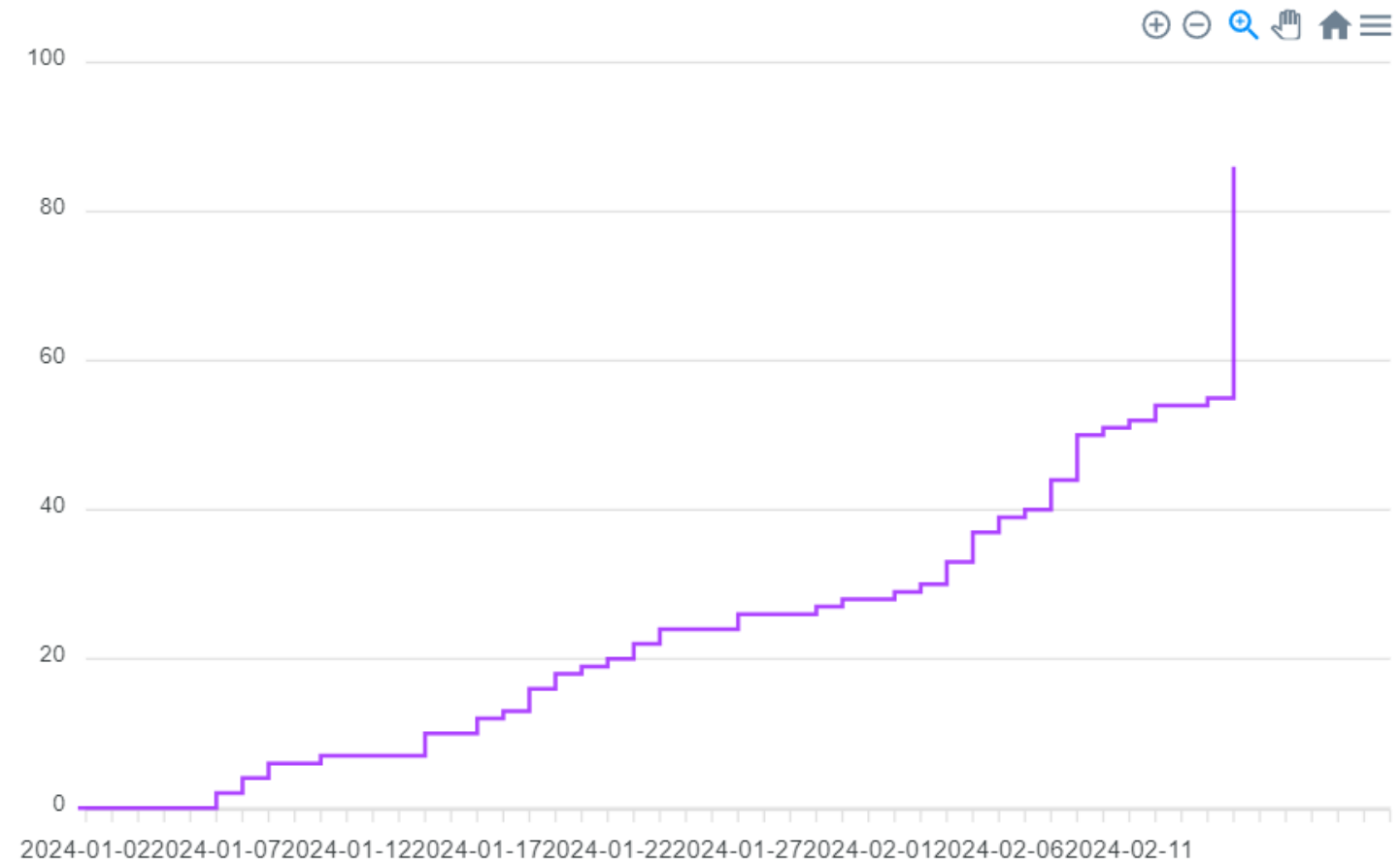
호버_시_갯수

나도_마스터가_됬어야

대시보드

- 누적 문제 갯수

365일의 걸음, 폰 문제



누적_문제_갯수

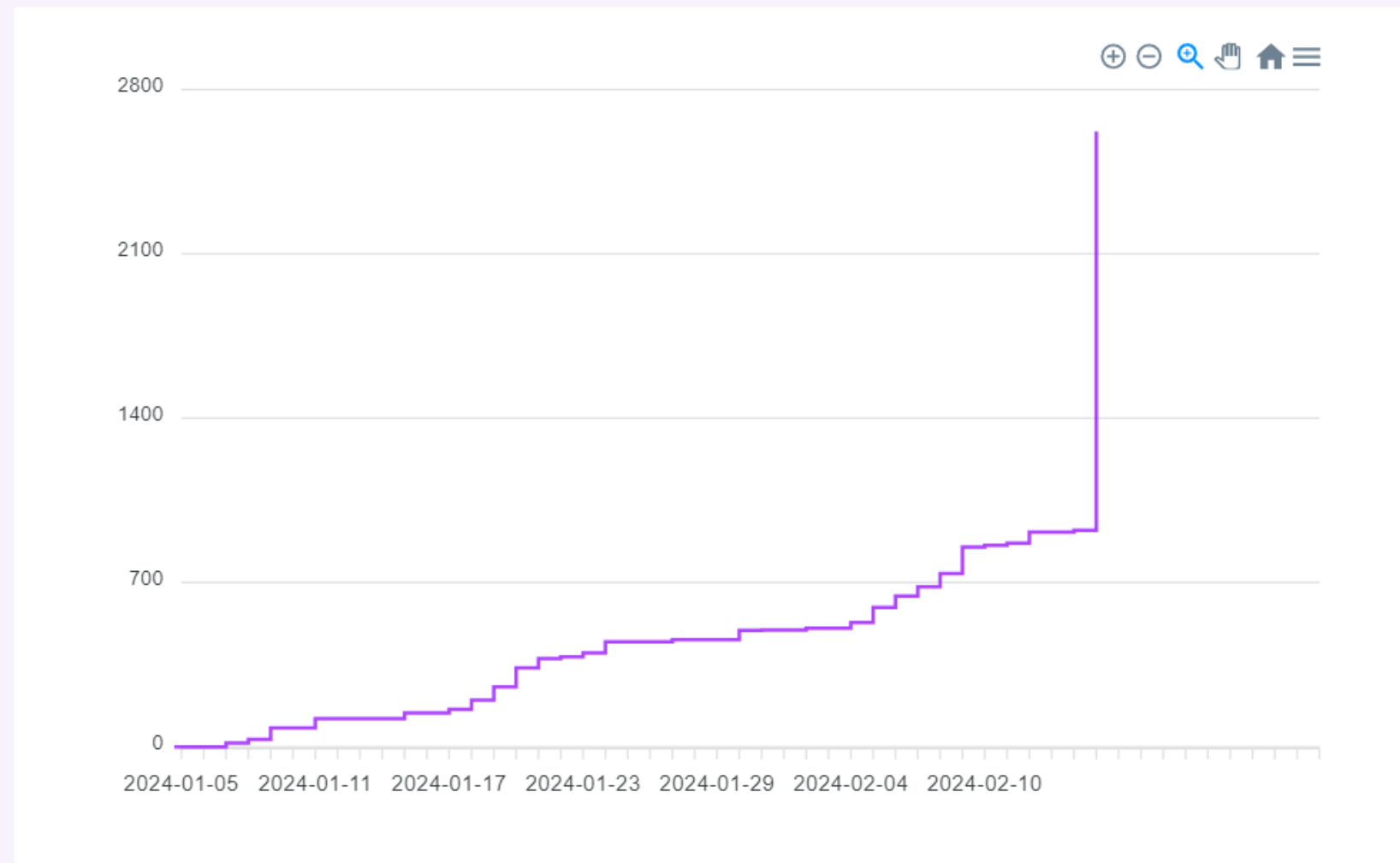
선형_그래프

원하는 기간_확대_가능

대시보드

- 누적 점수

365일의 걸음, 누적 점수



누적_점수 # 선형_그래프

발표_전날_벼락치기는_국룰이죠?

Community

ALGO;

나의정보 코드로그 커뮤니티 Q 유저검색



방 제목

코드

```
import java.util.*;

public class Main {
    public static void main(String[] args) {
        int a = 10;
        int b = 9;

        int c = a + b;

    }
}
```

고요한 코딩공토끼

흠 그거 그렇게 하는거 ㅇ

정열적인 코딩공

Community

Signaling Server node

```
import java.util.*;

public class Main {
    public static void main(String[] args) {
        int a = 10;
        int b = 9;

        int result = a +
    }
}
```

고요한 코딩공토끼

흠 그거 그러
정열적인 코딩공

I

Community

동시성, 작성자 표기

마크다운 형식

```
import java.util.*;

public class Main {
    public static void main(String[] args) {
        int a = 10;
        int b = 9;

        int result = a +
    }
}
```

고요한 코딩공토끼

흠 그거 그러
정열적인 코딩공

I

Community

동시성, 작성자 표기

마크다운 형식

```
import java.util.*;

public class Main {
    public static void main(String[] args) {
        int a = 10;
        int b = 9;

        int result = a +
    }
}
```

고요한 코딩공토끼

흠 그거 그러
정열적인 코딩공

I

Community

동시성, 작성자 표기

마크다운 형식

```
import java.util.*;

public class Main {
    public static void main(String[] args) {
        int a = 10;
        int b = 9;

        int result = a +
    }
}
```

고요한 코딩공토끼

흠 그거 그러

정열적인 코딩공

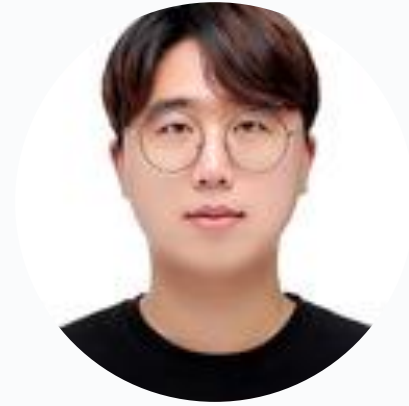
I

역할



엄예진
(팀장)

- API 설계 및 구현
- 협업툴 및 문서 관리
- 모델링 및 DB 설계



김수환
(BE 리더)

- 인프라 구축
- API 설계 및 관리
- 외부 API 관리



신유진
(BE)

- 모델링 및 DB 설계
- API 설계 및 구현
- Spring Security, oauth



김예지
(FE 리더)

- 인프라 구축
- UI/UX 설계 및 구현
- webRTC 구현



전소현
(UI/UX)

- UI/UX 설계 및 구현
- 와이어프레임
및 API 설계
- 데이터 스크래핑



정원종
(UI/UX)

- UI/UX 설계 및 구현
- 와이어프레임
및 API 설계
- 데이터 스크래핑

기술 스택

Management Tool



NOTION



MATTERMOST



JIRA

Infra



AMAZONAWS



NGINX



DOCKER



UBUNTU

Backend



SPRINGBOOT



SPRING SECURITY



MYSQL



SWAGGER

Front



REACT



TYPESCRIPT



SCSS



VITE

감사합니다