

Unified Job Search System

CONTENTS

01

Introduction

02

Web Crawling for Job Posts

03

Automated News Fetching

04

Trouble Shooting

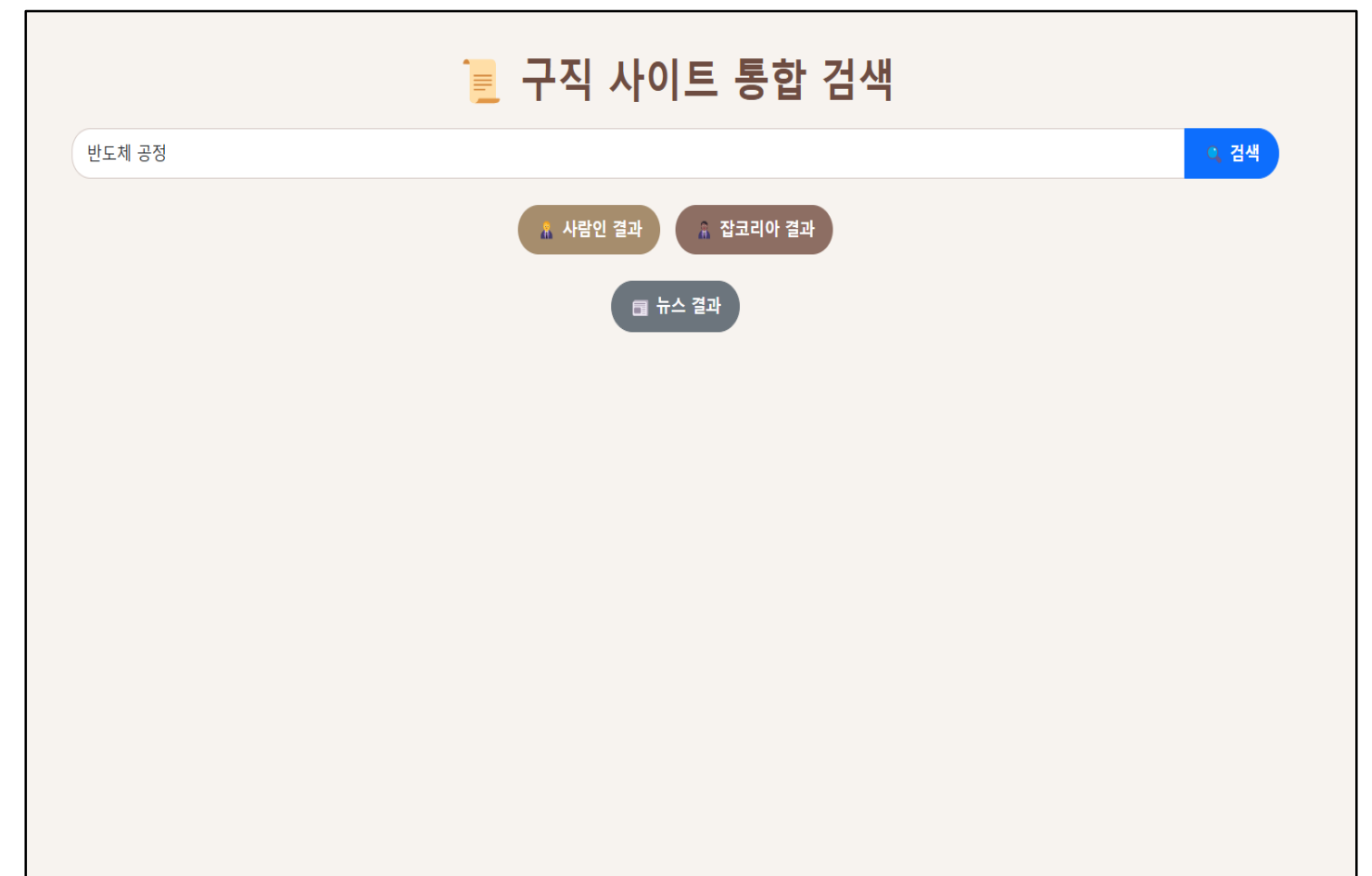
05

Summary

Introduction

취준생을 위한 구인구직 통합 시스템

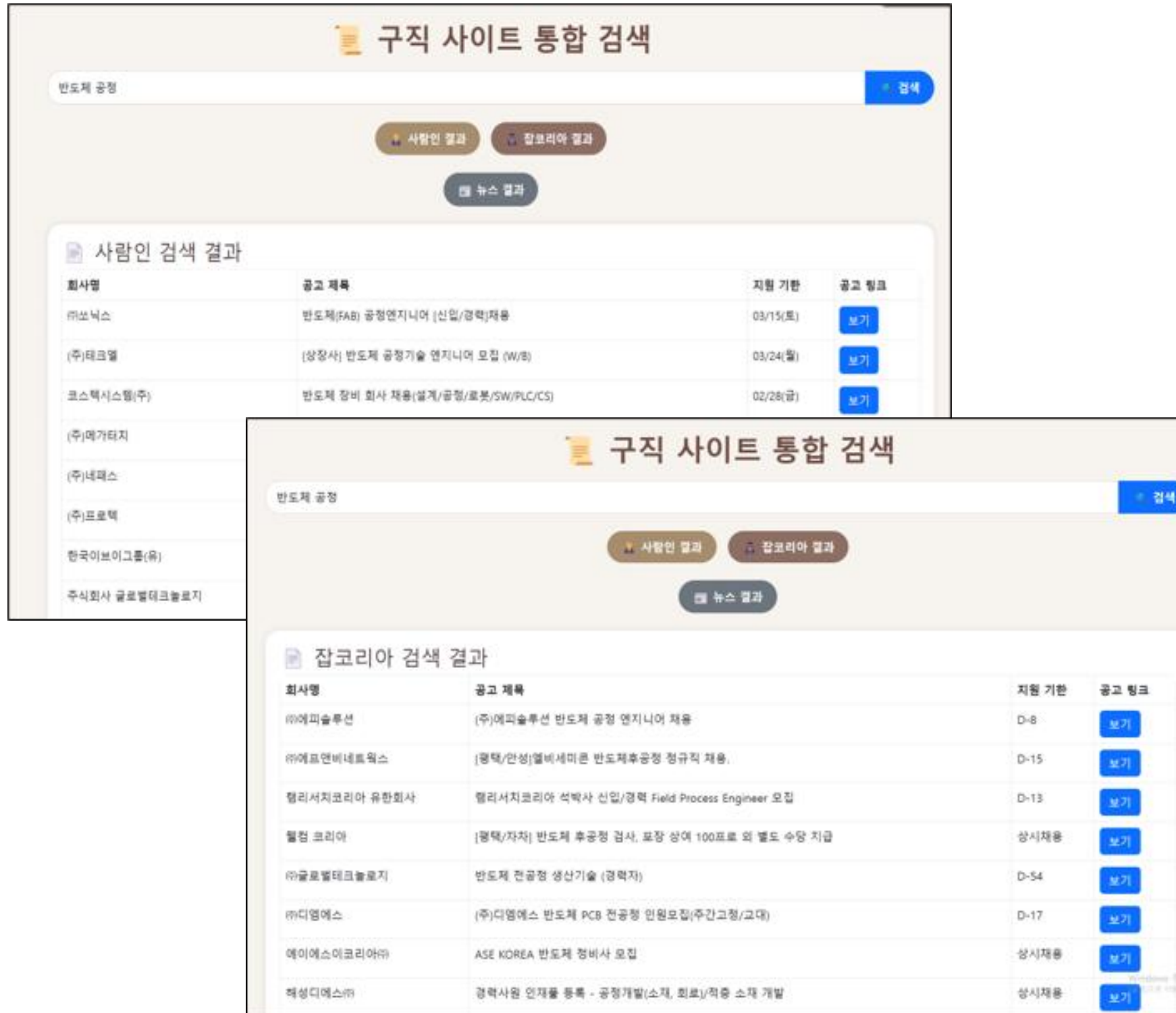
웹 크롤링을 활용한 자동 공고 수집과 지정된 시간마다
뉴스를 가져와 정리하는 자동화 시스템을 통해, 구직자들이
최신 채용 정보와 관련 뉴스를 한곳에서 효율적으로 탐색할
수 있도록 지원하는 통합 구인구직 플랫폼



Web Crawling for Job Posts

기능1_구인 공고 웹크롤링

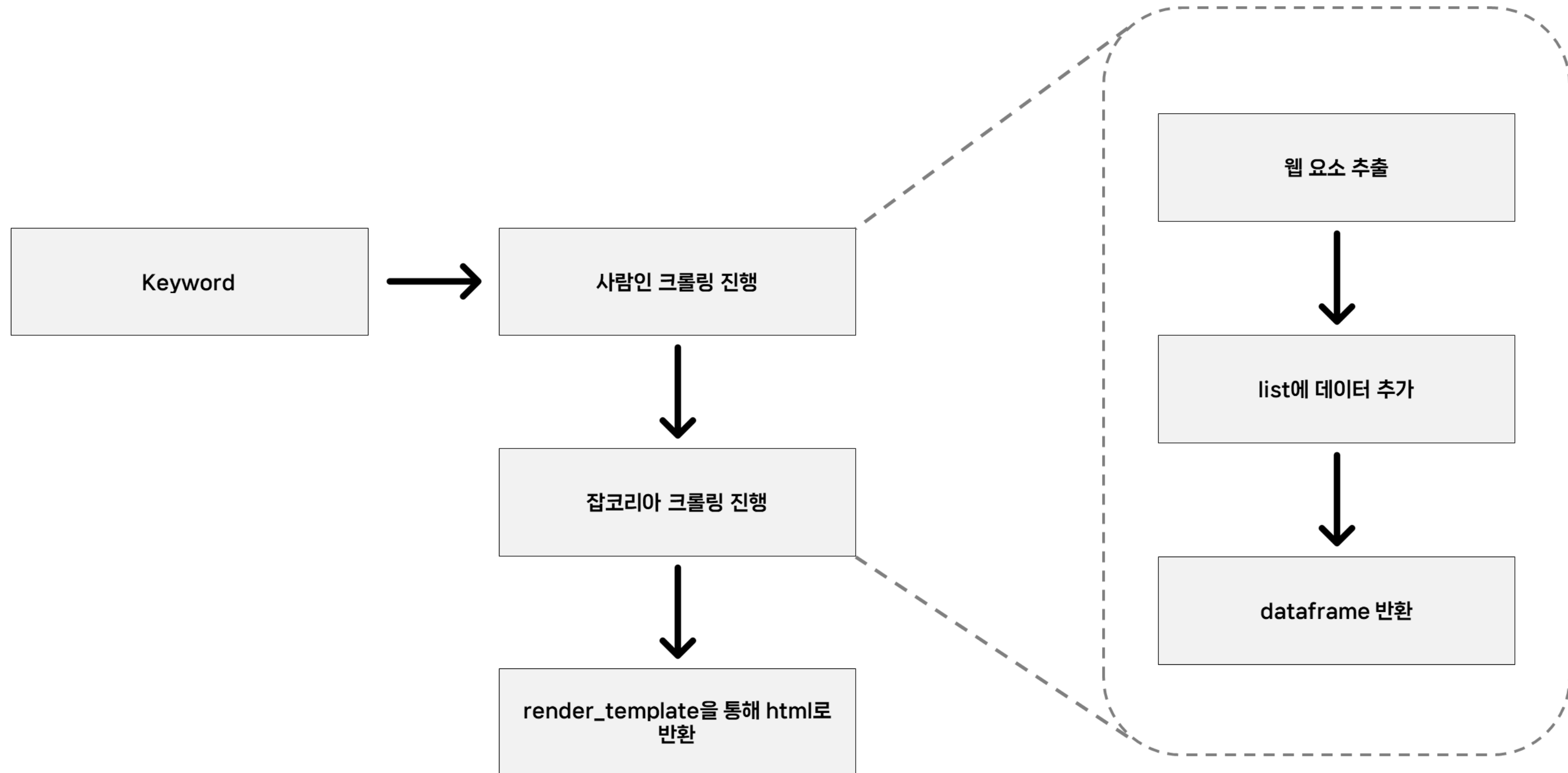
- ✓ 웹 크롤링 활용하여 특정 키워드의 구인 공고를 자동으로 수집하고 데이터프레임 형태로 저장.
- ✓ Flask 웹 서버를 활용해 수집된 뉴스 데이터를 HTML 페이지에서 시각적으로 제공.



회사명	공고 제목	지원 기한	공고 링크
리소닉스	반도체(FAB) 공정엔지니어 (신입/경력)채용	03/15(토)	보기
(주)태크엘	[상장사] 반도체 공정기술 엔지니어 모집 (W/B)	03/24(월)	보기
코스텍시스템(주)	반도체 장비 회사 채용(설계/공정/로봇/SW/PLC/CS)	02/28(금)	보기

회사명	공고 제목	지원 기한	공고 링크
(주)에피솔루션	(주)에피솔루션 반도체 공정 엔지니어 채용	D-8	보기
(주)에프엔비네트웍스	[광택/안성]엘비세미콘 반도체후공정 정규직 채용.	D-15	보기
텔리서치코리아 유한회사	텔리서치코리아 석박사 신입/경력 Field Process Engineer 모집	D-13	보기
월컴 코리아	[광택/자차] 반도체 후공정 검사, 포장 상여 100프로 외 별도 수당 지급	상시채용	보기
(주)글로벌테크놀로지	반도체 전공정 생산기술 (경력자)	D-54	보기
(주)디엠에스	(주)디엠에스 반도체 PCB 전공정 인원모집(주간고정/고대)	D-17	보기
에이에스에이코리아(주)	ASE KOREA 반도체 정비사 모집	상시채용	보기
해성디에스(주)	경력사원 인재를 등록 - 공정개발(소재, 회로)/직종 소재 개발	상시채용	보기

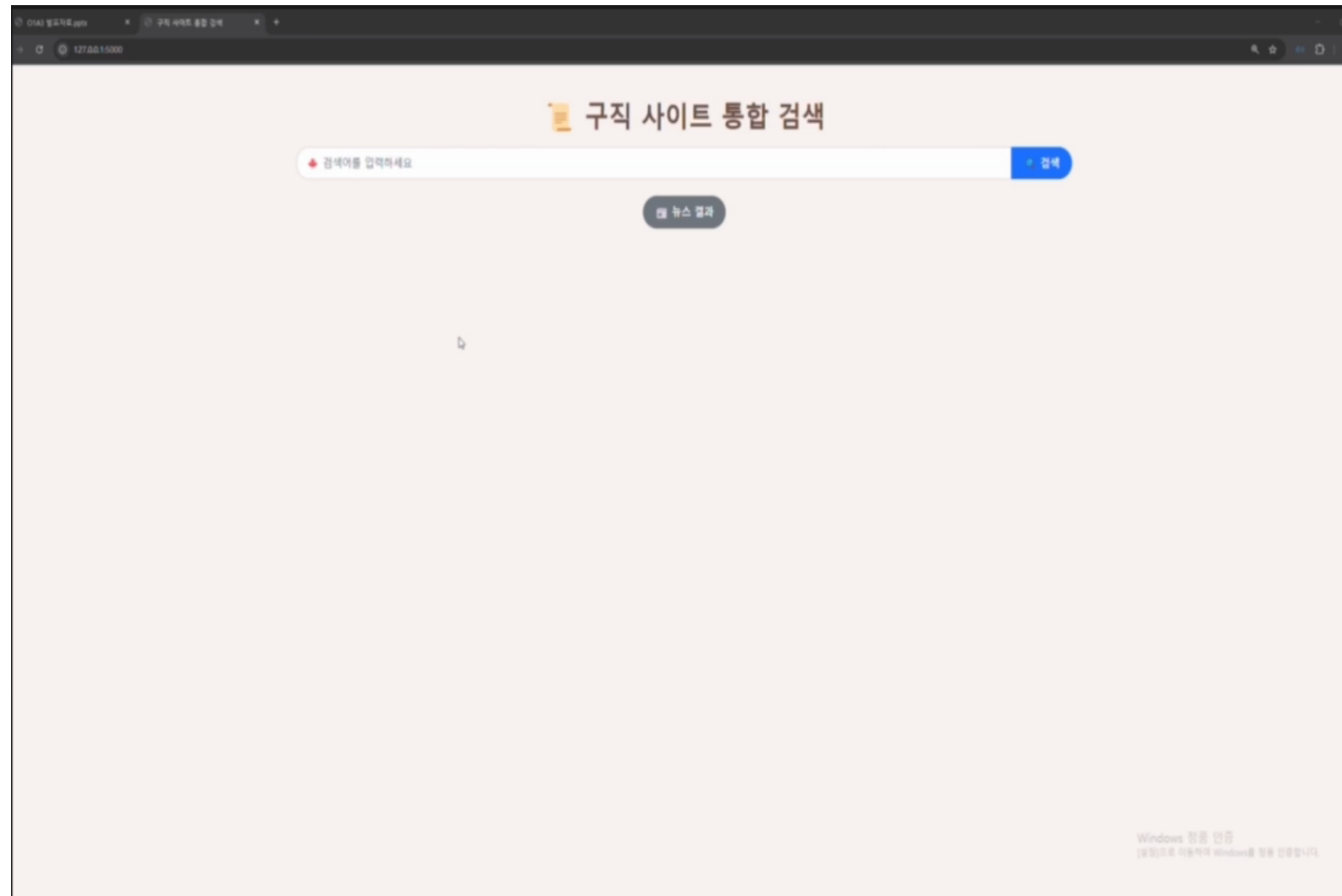
Flow Chart



Web Crawling for Job Posts



기능1_구인공고 웹크롤링_시연 영상



Automated News Fetching



기능2_뉴스 수집 자동화

✓ 네이버 뉴스 API를 활용하여 특정 키워드의 최신

뉴스를 자동으로 수집하고 CSV 파일로 저장.

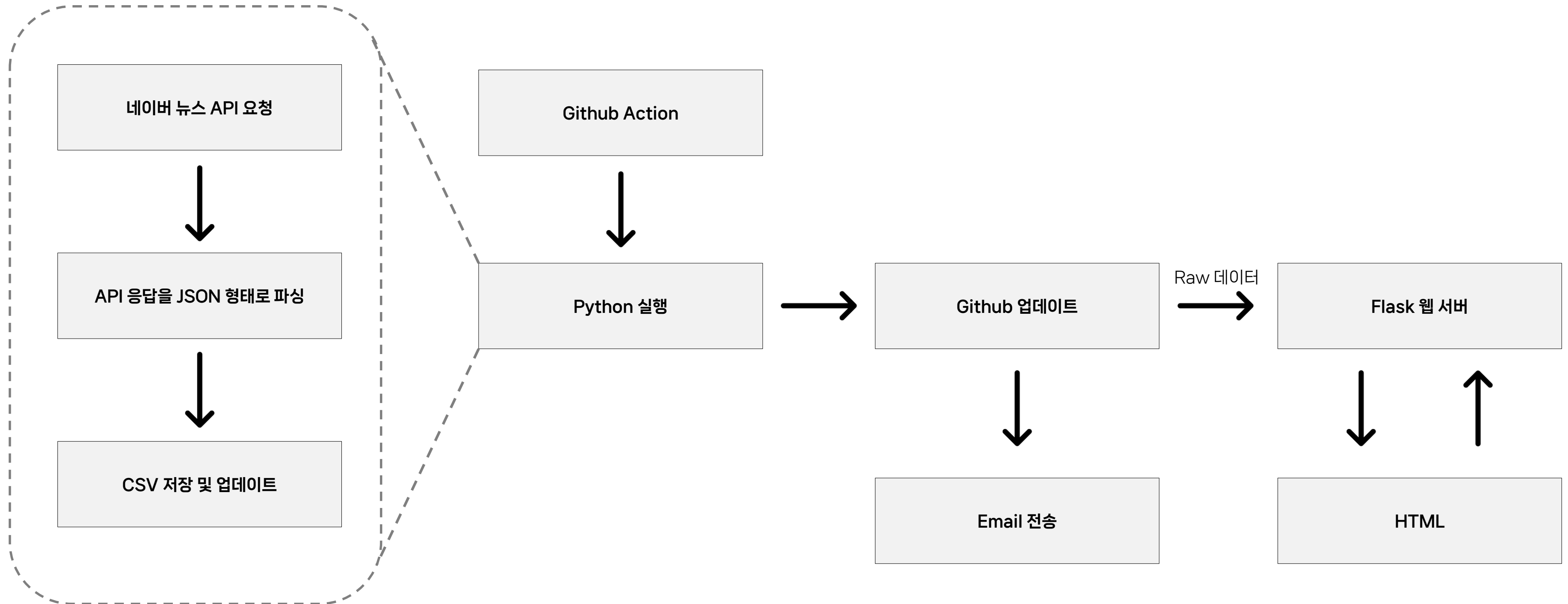
✓ GitHub Actions를 사용해 주기적으로

실행하여 최신 뉴스 데이터를 유지.

✓ Flask 웹 서버를 활용해 수집된 뉴스 데이터를

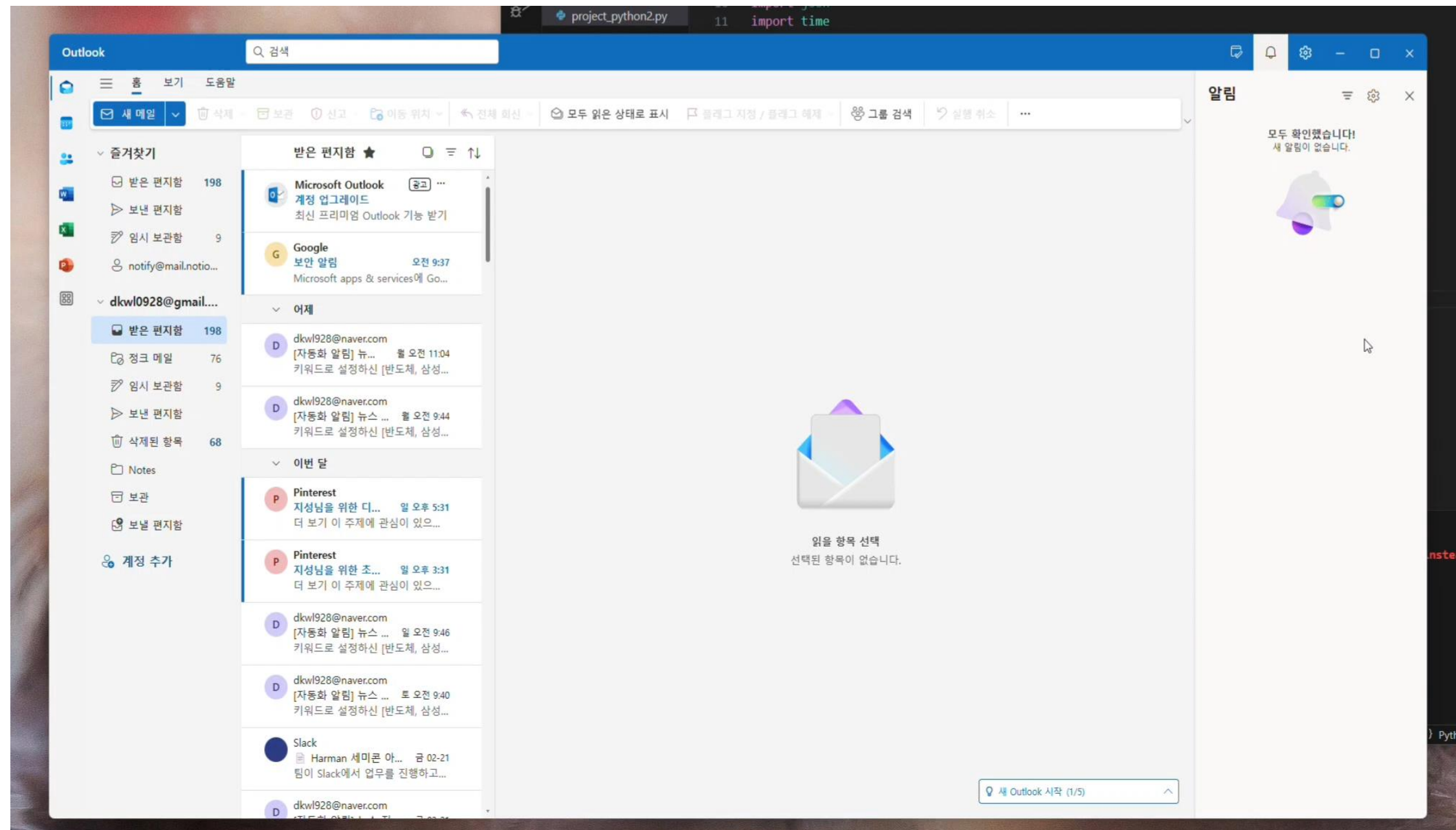
HTML 페이지에서 시각적으로 제공.

Flow Chart



Automated News Fetching

기능2_뉴스자동화_시연 영상



Trouble Shooting

문제 상황

```
def crawl_saramin(keyword):
    print(f"[사람인] '{keyword}' 검색 시작...")
    service = Service(ChromeDriverManager().install())
    driver = wb.Chrome(service=service)
    try:
        driver.get("https://www.saramin.co.kr")
        search_box = driver.find_element(By.CSS_SELECTOR, ".search #btn_search")
        search_box.click()
        search_input = driver.find_element(By.CSS_SELECTOR, ".option .key")
        search_input.send_keys(keyword)
        search_input.send_keys(Keys.ENTER)
        # 채용정보 탭 클릭 (찾지 못하면 예외 처리)
        try:
            job_info_tab = driver.find_element(By.CSS_SELECTOR, ".tab_search_result li+li .tab")
            job_info_tab.click()
        except Exception as e:
            print("채용정보 탭을 찾지 못했습니다.")
        job_title_list, job_date_list, corp_name_list, job_link_list = [], [], [], []
        page_number = 1
```

① 구현하고자 했던 부분

웹 크롤링 데이터를 빠르게 취합하려 시간 지연을 사용하지 않고
웹 크롤링 진행 요소를 찾을 수 없음

② 발생한 이슈

크롤링 시 웹사이트 로딩으로 인한 선택한 CSS 선택자로 요소를
찾지 못해 오류 발생

해결

```
def crawl_saramin(keyword):
    print(f"[사람인] '{keyword}' 검색 시작...")
    service = Service(ChromeDriverManager().install())
    driver = wb.Chrome(service=service)
    try:
        driver.get("https://www.saramin.co.kr")
        time.sleep(3)
        search_box = driver.find_element(By.CSS_SELECTOR, ".search #btn_search")
        search_box.click()
        time.sleep(1)
        search_input = driver.find_element(By.CSS_SELECTOR, ".option .key")
        search_input.send_keys(keyword)
        search_input.send_keys(Keys.ENTER)
        time.sleep(3)
        # 채용정보 탭 클릭 (찾지 못하면 예외 처리)
        try:
            job_info_tab = driver.find_element(By.CSS_SELECTOR, ".tab_search_result li+li .tab")
            job_info_tab.click()
            time.sleep(3)
```

③ 해결 방법

CSS 선택자를 업데이트 time.sleep() 사용해 로딩 시간의 여유를
적용시켜 데이터를 안정적으로 취합 진행

Trouble Shooting

문제 상황

```
8 GITHUB_CSV_URL = "https://raw.githubusercontent.com/kjs928/
9
10 response = requests.get(GITHUB_CSV_URL)
11 response.raise_for_status()
12
13 df = pd.read_csv(io.StringIO(response.text))
14
15 # 날짜 리스트 생성
16 df_date_list = list(df.drop_duplicates(["date"])["date"])
17
18 # 날짜 리스트에 한달 전 날짜까지만 저장
19 if len(df_date_list) > 30:
20     df_date_list = df_date_list[0:30]
21
22 # 날짜별 데이터 프레임 생성
23 for i in df_date_list:
24     globals()['df_' + i] = df.loc[df["date"]==i]
25
26 # 키워드별 데이터 프레임 생성
27 for i in df["keyword"]:
28     globals()['df_' + i] = df.loc[df["keyword"]==i]
29
30 print(globals()['df_2025-02-20'])
31
32 json = json.dumps(df, ensure_ascii=False, indent=2)
33
34
```

① 구현하고자 했던 부분

CSV 파일을 받아 날짜별 및 키워드별로 개별 데이터프레임을 생성한 후, 이를 render template을 통해 HTML로 전달하려 함.

② 발생한 이슈

날짜와 키워드별로 데이터프레임을 생성하면 데이터프레임의 수가 너무 많아져서 파일 관리가 어려워지고 코드가 복잡해짐.

해결

```
176
177 GITHUB_CSV_URL = "https://raw.githubusercontent.com/kjs928/HARMAN_Pytho
178 response = requests.get(GITHUB_CSV_URL)
179 response.raise_for_status()
180 df = pd.read_csv(io.StringIO(response.text))
181 news_json = df.to_dict(orient="records")
182
183 # 뉴스 가져 오기
184 @app.route('/news')
185 def news():
186     return render_template("home.html", news_data=news_json)
187
188 @app.route('/date')
189 def date():
190     return render_template("date.html", news_data=news_json)
191
192 @app.route('/keyword')
193 def keyword():
194     return render_template("keyword.html", news_data=news_json)
195
196 if __name__ == "__main__":
197     app.run(debug=True)
```

③ 해결 방법

자바스크립트와 JSON을 활용하여 HTML 내부에서 데이터를 필터링하도록 변경. 이로 인해 코드가 간결해지고 수정이 용이해짐.

Project Summary

프로젝트 목표

취준생들을 위한 취업공고와 뉴스를 한 곳에서
편리하게 볼 수 있는 통합 시스템 구현

구현 기능

1. 웹크롤링을 활용한 공고 데이터 가져오기
2. API, github action을 활용한 뉴스 자료 가져오기

사용 기술

1. Flask
2. BeautifulSoup & Selenium
3. Requests
4. Pandas
5. Schedule & GitHub Actions
6. HTML
7. Email (smtplib)

Team Members

강x지



- 뉴스 자동화 기능 구현
- Flask 서버 구현

고x식



- 잡코리아 크롤링 구현
- Flask 서버 구현

강x성



- 발표
- 뉴스 자동화 기능 구현
- Flask 서버 구현

구x민



- 사람인 크롤링 구현
- Flask 서버 구현

THANK YOU