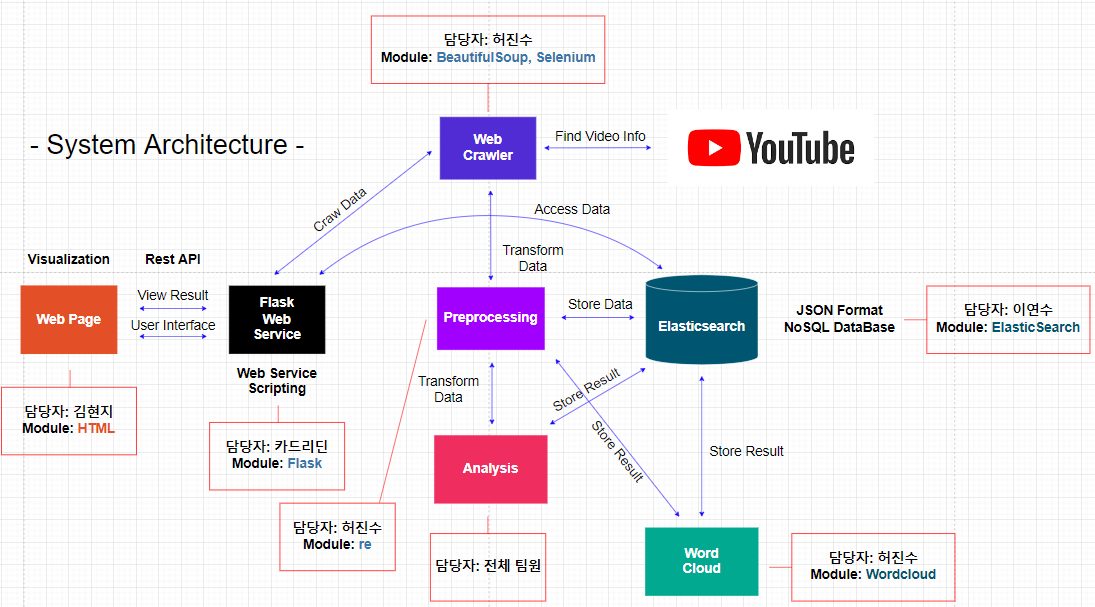
**프로젝트명:<유튜브 댓글 긍정/부정 분석과 워드 클라우드 표현>**

1. **개요**

요즘 유투브에서 영상을 업로드해 수익을 창출하는 유투버들이 증가하고 있는 추세이다. 그리고 영상 조회수를 높이기 위해 유투버가 시청자들의 요구 사항 및 관심도를 파악하는 것이 필수적이다. 하지만, 유투버가 많은 댓글들을 모두 보기가 어렵기 때문에 이를 해결하기 위해 영상의 댓글을 분석해 영상에 대한 반응, 관심도 등을 워드 클라우드를 통해 시각화 하는 서비스를 만들게 되었다. 이 서비스가 실용화된다면 유투버 뿐만 아니라 일반 사용자에게도 도움이 될 것으로 기대하고 있다. 유투브는 일부 영상에 대해 제제를 하지만, 댓글에는 제제가 없기 때문에 사용자가 불필요한 댓글을 보는 경우가 많다. 그래서, 이 서비스를 이용한다면 사용자가 전반적인 댓글을 파악할 수 있어 댓글창을 볼 것인지 판단할 수 있도록 해 줄 것이다.

이 프로젝트는 Github를 통해서 팀원들과 같이 협업하였다. Beautifulsoup과 selenium을 사용하여 유튜브 댓글을 크롤링 하고 Keras konlpy model를 사용하여 댓글을 분석한다. 그리고 그 결과를 wordcloud로 시각화하여 결과와 함께 elasticseach를 이용해 저장하였다. flaskwebservice를 이용하여 webpage에 결과를 보여주는 방식으로 구현하였다.

1. **시스템 구조**



※ 설명

**사용한 언어**: HTML, CSS, JavaScript, Python

**사용한 Tool**: Flask, Elasticsearch, Docker

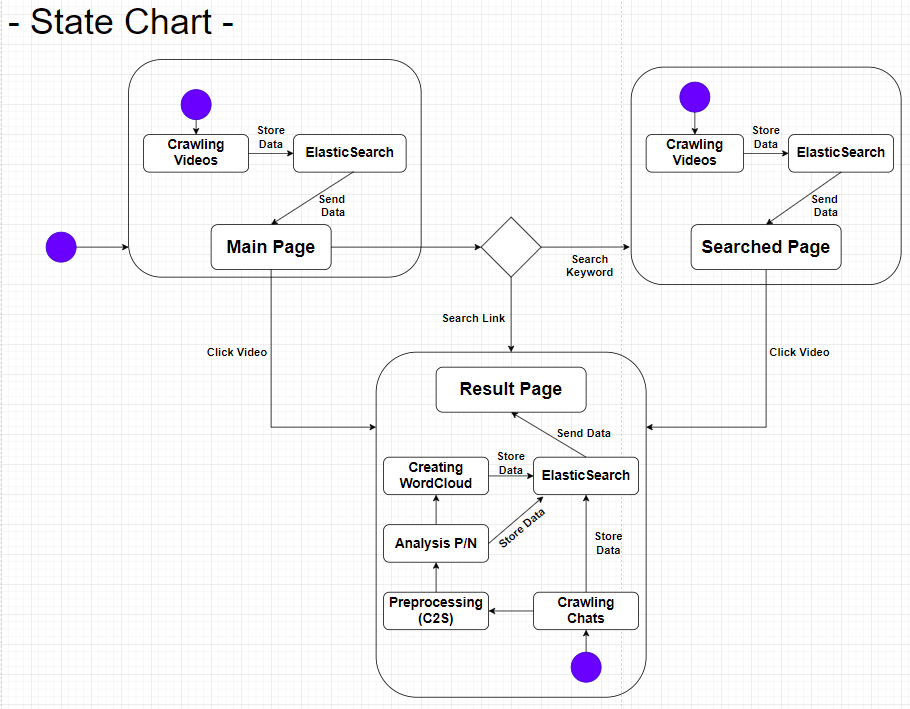
**Version 관리**: Github

**프로젝트 일정이나 회의 관리**: Notion

**개발 환경**: Visual Studio Code, PyCharm

시스템을 개발할 때 실행환경으로 VMware를 설치하여 정상적으로 작동하는지 확인하였고, Docker는 사용할 사람만 Container를 생성하여 시스템이 작동되는지 확인했다.

전반적인 Component 간의 상호작용은 위 그림을 참고하면 된다.



※ 설명

- 처음 시스템을 실행할 때 유튜브의 메인 페이지에 있는 영상을 10개 크롤링한다. 이때 영상의 이미지, 제목, 주소, 조회수, 좋아요 수 데이터를 가져오는데 만약 크롤링이 안되면 None 값으로 둔다. 크롤링이 완료되면 Elasticsearch에 데이터를 저장하고, 저장한 데이터를 Elasticsearch에서 가져와서 Main 페이지로 보낸다.

- Main 페이지에서는 사용자가 원하는 키워드(단어) 혹은 동영상 주소(링크)를 검색할 수 있고, 크롤링으로 가져온 10개의 동영상에 대한 데이터를 카드 형태로 나열하여 보여준다. 동영상에 대한 각 카드를 사용자가 클릭하여 영상이 긍정적인지, 부정적인지 확인할 수 있는 결과 페이지로 넘어갈 수 있다.

- 사용자가 키워드(단어)로 검색을 하는 경우, 크롤러를 실행하여 입력된 키워드에 대한 10개의 동영상 정보(이미지, 제목, 주소, 조회수, 좋아요 수)를 가져오는데 만약 해당 정보가 없으면 None으로 둔다. 크롤링이 완료되면 Elasticsearch에 데이터를 저장하고, 저장한 데이터를 Elasticsearch에서 가져와서 검색 결과 페이지로 보낸다.

- 검색 결과 페이지에서는 사용자에게 크롤링으로 가져온 10개의 동영상에 대한 데이터를 카드 형태로 나열하여 보여주고, 사용자는 각 카드를 클릭하여 영상이 긍정적인지, 부정적인지 확인할 수 있는 결과 페이지로 넘어갈 수 있다.

- 사용자가 키워드가 아닌 동영상의 링크로 검색을 하는 경우, 크롤러를 실행하여 해당 동영상의 댓글을 가져오고, 댓글 데이터를 문장 단위로 나누어서 데이터를 가공한다. 그 후에, 해당 영상이 긍정적인지, 부정적인지 분석하고 긍정/부정 비율이 더 높은 쪽의 문장 데이터를 가져와서 Word Cloud를 만든다. 그리고 분석한 결과와 생성한 Word Cloud 이미지를 Elasticsearch에 저장한다. 모든 작업이 끝나면 해당 동영상에 대한 데이터를 Elasticsearch에서 불러와서 결과 페이지로 보낸다.

- 사용자가 영상 카드를 클릭한 경우, 사용자가 동영상 링크를 검색했을 때 시스템에서 처리하는 작업과 동일하다.

- 결과 페이지에서는 해당 영상에 대한 긍정/부정 분석 결과 데이터와 Word Cloud 데이터를 영상과 함께 사용자에게 보여준다. 구체적으로는 가장 많이 나온 긍정적인 단어 Top 3와 부정적인 단어 Top 3, 해당 영상과 영상에 대한 Word Cloud 이미지, 긍정/부정 비율을 보여준다.

1. **기본/추가 기능 설계**

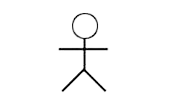
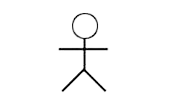
Web functions from the user’s perspective: UML - Use Case Diagram

User

(Actor)

extend

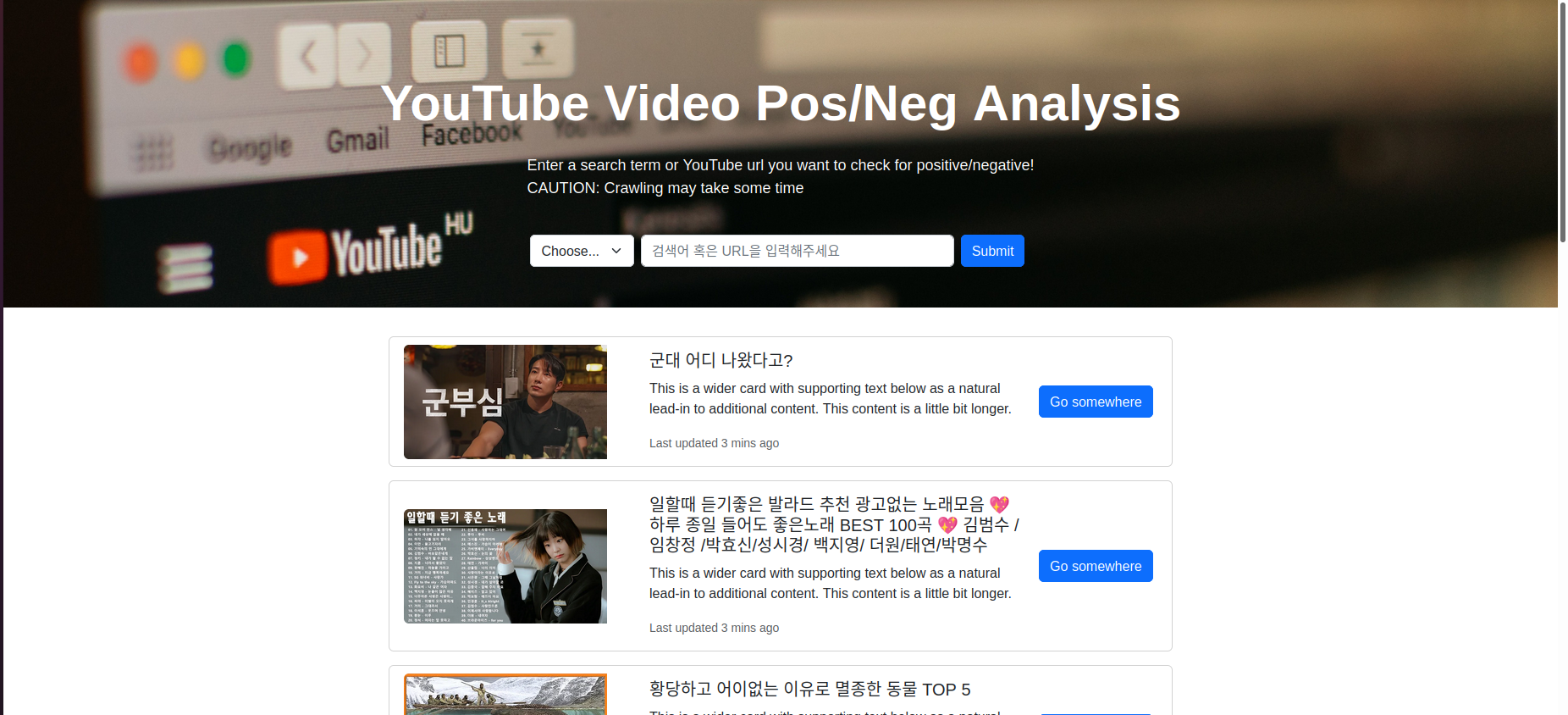
extend



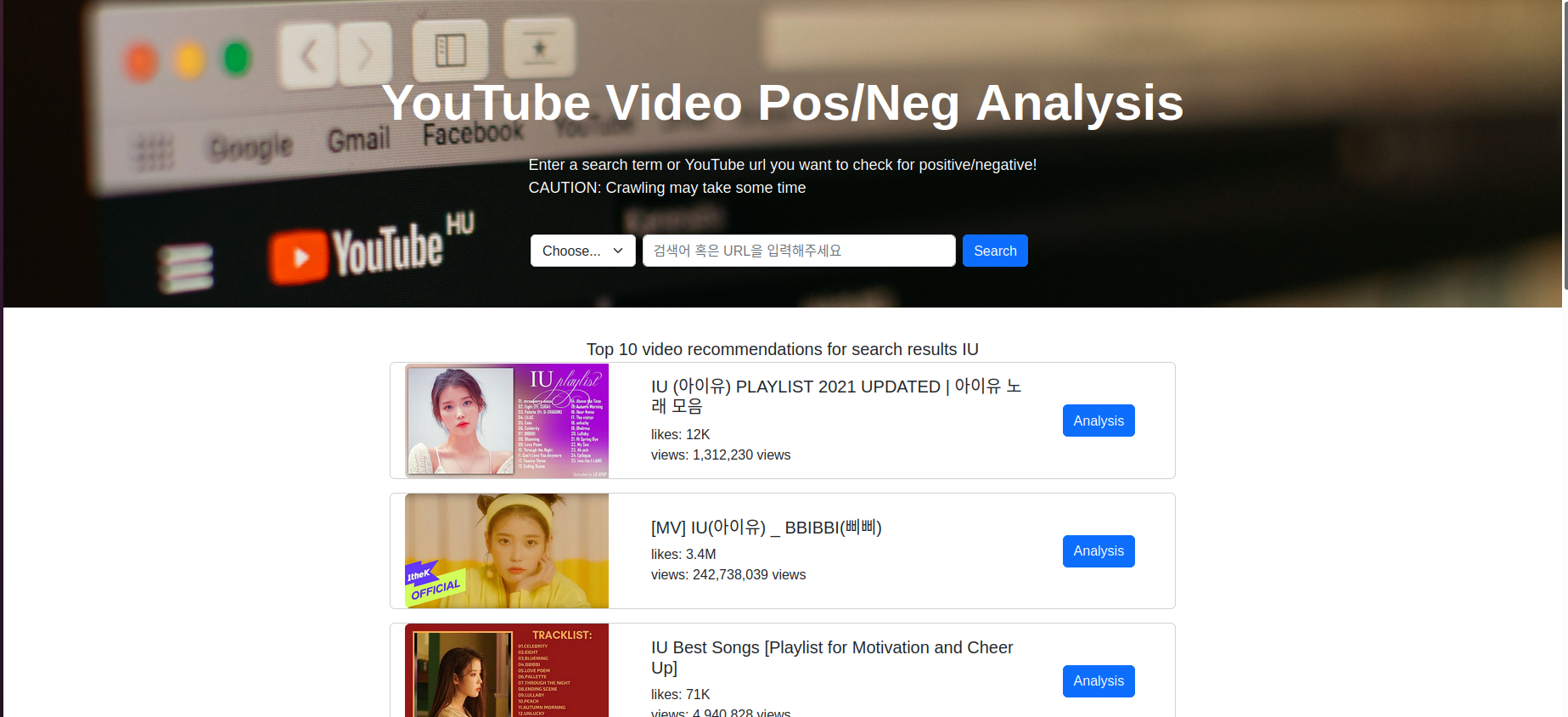
Server

System

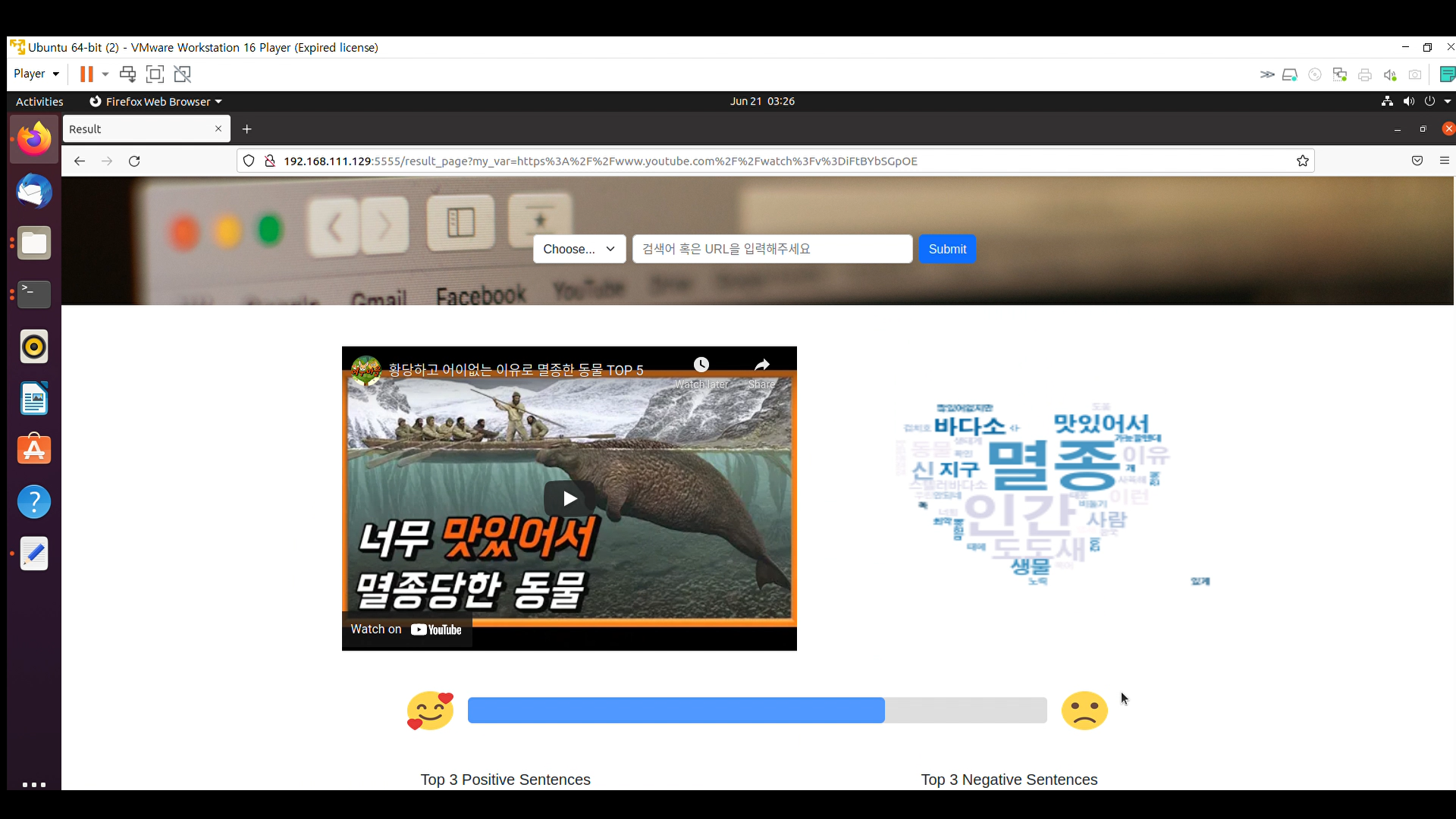
1. Web home page이 열 때 selenium & beautifulsoup를 통해서 youtube 실시간 recommendation list에 10개 동영상의 img, url, 제목, 좋아요 수, 조회 수를 crawling 해 온다.
2. 1번에 crawling한 정보를 url ID, img, 제목, 좋아요 수, 조회수, 데이터를 묶음으로 Elasticsearch를 통해 indices “home\_data” index에 저장한다.
3. Elasticsearch indices “home\_data” index에 해당 url로 저장된 데이터를 home page에 보내고 web page에 10 개 영상의 img, 제목, 좋아요 수, 조회 수를 각 card로 묶어서 recommendation list로 보여준다.
4. 사용자에게 보여지는 web page에0 사용자로부터 url로 검색하는지 단어로 검색하는 설정한다.
5. 4번에 단어로 검색하는 option를 선택하고 단어를 입력 받으면, selenium & beautiful soup library들 통해서 해당 단어로 youtube에 나오는 10개 영상들의 영상0, url, img, 제목, 좋아요 수, 조회수를 crawling 해온다.
   1. 5번에 crawling한 정보를 단어 ID, 영상, url, img, 제목, 좋아요, 조회 수, 데이터를 묶음으로 Elasticsearch를 통해 indices “search\_data” index에 저장한다.
   2. Elasticsearch indices “search\_data” index에 해당 단어로 저장된 데이터를 search page에 보내고 web page에 10 개 영상의 img, 제목, 좋아요 수, 조회 수를 각 card로 묶어서 word 검색 결과 list에 선택할 수 있게 보여준다.
6. 4번에 url로 검색하는 option를 선택하고 url로 입력 받거나, home page recommendation videos list에 video를 선택하거나, search page 단어 검색 결과 list에 분석하고 싶은 video를 선택하면 해당 video의 url 통해서 영상 댓글들 crawling 한다.
7. 6번에 crawling 한 데이터를 url ID, 댓글 Elasticsearch “com\_data” index에 저장한다.
8. Elasticsearch “com\_data” index에 해당 url 데이터를 preprocessing class를 통해서 숫자, 영어 문자, 특수문자, emoticons를 지운다. Preprocessing 된 데이터를 학습시킨 konlpy, nltk, Keras model에 댓글 하나하나 돌리고 감정 분석하고 positive와 negative 댓글들 따로 분리하고 전제 댓글 수, positive 댓글 퍼센트, negative 댓글 퍼센트까지 계산한다.
9. 8번에 대다수 댓글들 positive인 경우 positive 댓글 list를 통해서 wordcloud를 만들고 대다수에 댓글들 negative이면 negative 댓글 list를 통해서 wordcloud를 만든다. 이때 pos/neg list를 통해서 만드는 것에 따라서 wordcloud font 색깔 다르게 나온다.
10. 전제 댓글 수, positive 댓글 present, pos/neg 댓글 list와 wordcloud를 elasticsearch result page에 보내도록 “result\_data” index에 저장한다.
11. **결과**

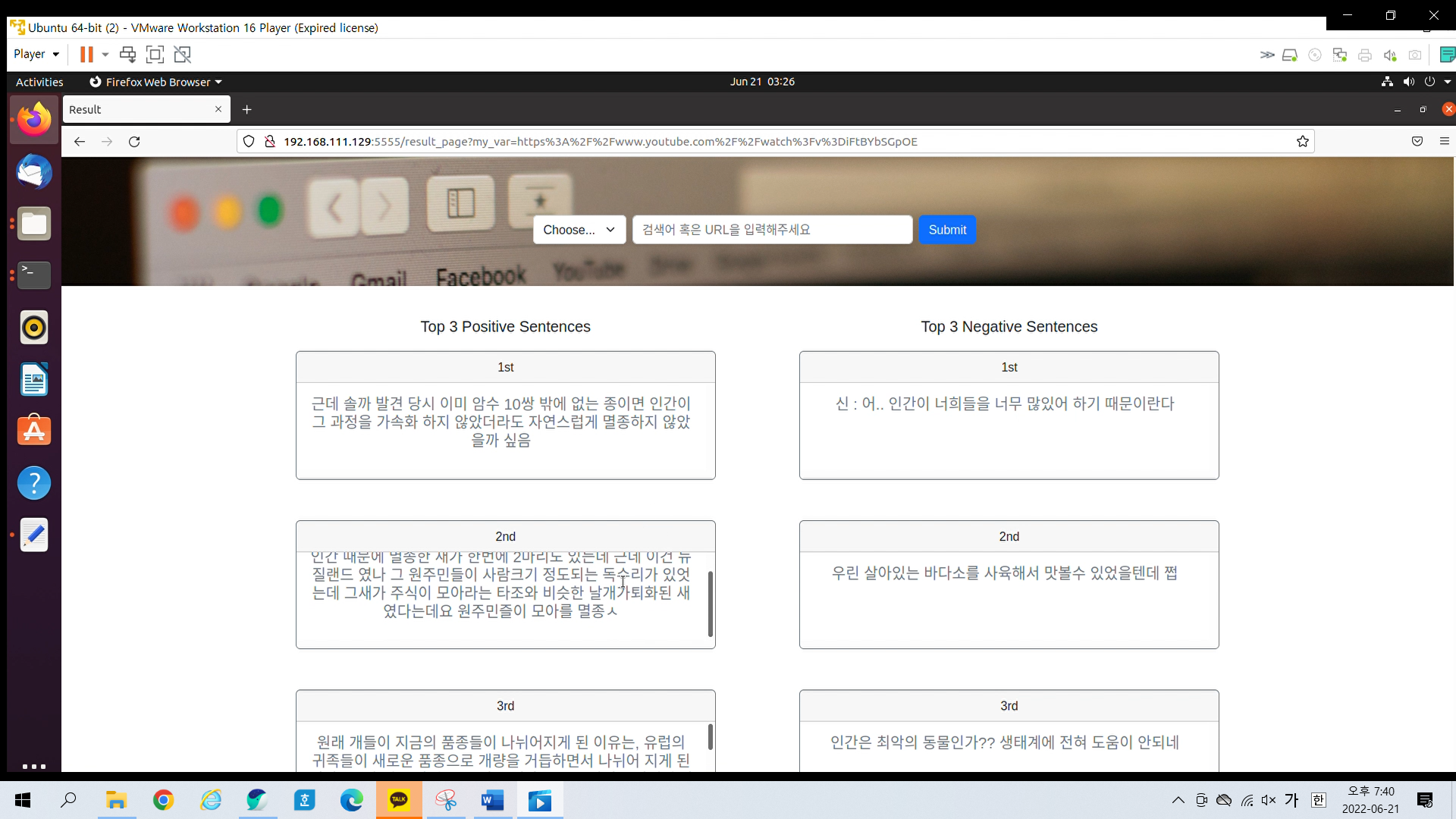


* url이나 검색어를 입력하여 검색할 수 있는 입력창과 실시간 급상승 동영상과 댓글을 크롤링하여 그 list를 보여준다. 여기서 result 버튼을 누르면 실시간 급상승 동영상의 댓글 분석 결과 페이지를 보여준다.



* 검색어를 입력했을 경우 그 검색어와 관련된 동영상을 크롤링하여 그 list를 보여준다. 여기서 result 버튼을 누르면 다음 창으로 넘어간다





* result 버튼을 누르거나 url로 검색할 경우 나오는 댓글 분석 결과 화면이다. 크롤링하여 분석한 댓글을 토대로 긍정과 부정의 정도를 쉽게 파악할 수 있도록 bar를 이용하여 표현하였다. 또한, 긍정적인 댓글 3개와 부정적인 댓글 3개를 노출시켜 사용자가 영상에 대한 전반적인 반응을 한눈에 파악하기 쉽도록 하였다. 마지막으로 그 문장을 토대로 단어들을 추출해 wordcloud를 시각화하여 전반적인 내용을 볼 수 있도록 만들었다.

1. **느낀점**

- 김현지 : 한 학기 동안 Linux, Flask, Elasticsearch 등 새로운 환경에서 프로젝트를 처음부터 끝까지 진행해볼 수 있어서 뜻깊었습니다. 평소에 혼자서 공부한다면 접하기 쉽지 않을 기능들을, 프로젝트를 하면서 팀원들과 새로운 기능을 추가하고 계속 시도해보면서 많이 배우고 성장할 수 있었습니다. 낯선 환경에서의 코딩이듯이, 에러도 많고 막힐 때도 많았지만, 함께 에러를 수정하고 코드를 분석할 수 있어서 좋았습니다. 특히, github는 개발자 협업을 위해 많이 사용되는 것을 알고 있음에도, 꾸준히 사용하거나 활용하는 것이 어려웠는데, 팀프로젝트를 하면서 github를 통해 협업하는 방법과 과정을 직접 경험할 수 있어서 좋았습니다. open source 라는 단어만 들어보고, 정확히 무엇인지, 내가 어떻게 활용할 수 있을지에 대한 생각은 미처 하지 못했었는데, 이번 오픈소스프로그래밍 수업을 수강하면서 '나도 이제 오픈소스를 활용할 수 있겠다' 라는 자신감을 얻을 수 있었습니다. 한 학기가 짧게 느껴질 정도로 재밌고 의미있었던 프로젝트였습니다.

- 이연수 : VM ware RAM 상으로 오류 뜰 때가 많아서 작동시키는데 시간을 많이 투자해야 했다. 또한, 긍정 댓글과 부정 댓글을 분석하는 방법을 찾을 때 속도가 느린 코드들이 많아 빠른 방법을 찾는데 많은 어려움이 있었다. 하지만, 그 과정에서 다양한 방법을 시도해보고 풍부한 지식을 쌓을 수 있는 좋은 경험이었던 것 같다.

- 카드리딘 : 팀원들끼리 communication이 잘 돼서 문제 생길 때 마다 빨리 해결하게 돼서 좋았다. 그리고 이번에는 VM Ubuntu 환경에 프로젝트를 하면서 새로운 것 많이 알게 됐고 실제로 RAM memory 중요성을 느껴 봤다.

- 허진수 : 여러 사람이 하나의 팀 프로젝트에 참여하여 서로 다른 작업 환경에서 코드를 작성하고 Git에 업데이트하니까 version 관리가 정말 힘들었다. 또, Docker를 사용하여 시스템을 실행했을 때와 VMware에서 시스템을 실행했을 때가 너무 달라서 환경 설정하기가 무척 까다로웠다. 그리고 코드를 작성할 때마다 정상적으로 실행되는지 확인하는 시간이 오래 걸려서 개발 기간이 예상보다 더 오래 걸린 것 같다.