창직종합설계프로젝트2

최종결과보고서



4조 B311184 임진수

B311213 조재성

B411145 이성현

B411103 백상욱

**1. 개요**

**1.1 프로젝트 명**

CODEME (빅 데이터를 이용하여 알고리즘 학습을 도와주는 프로그래밍 문제 추천 시스템)

**1.2 개발 동기**

4차 산업혁명 시대에 접어들면서 컴퓨터공학과의 위상이 높아지는 가운데 알고리즘 교육의 중요성이 대두되고 있습니다. 알고리즘 학습에 있어서 자신의 실력이 어느 정도인지 앞으로 어떤 문제를 풀어야 하는지 가이드라인을 제시하고 어떤 분야의 알고리즘에 취약한지 파악할 수 있게 하는 서비스를 제공하고자 하였다.

**1.3 목적**

알고리즘 학습을 목표로 하는 모든 사용자들의 실력 향상을 목표로 하지만 주된 목표는 취업을 앞두고 SW역량테스트(코딩테스트)를 준비하는 사용자들이 이 서비스를 이용하여 효율적으로 알고리즘 학습을 하는 것이다.

**1.4 요약**

백준 온라인 저지 사이트의 문제 풀이 정보를 참고하여 사이트 이용자의 문제 풀이 데이터를 데이터베이스에 저장한다. 데이터를 ELO rating 시스템 수식에 대입하여 자신의 rating을 측정할 수 있으며 이를 바탕으로 본인에게 맞는 수준의 백준 온라인 저지 사이트의 알고리즘 문제를 추천해주고 취약한 알고리즘 파트를 분석하여 제시해준다.

**1.5 개발 환경**

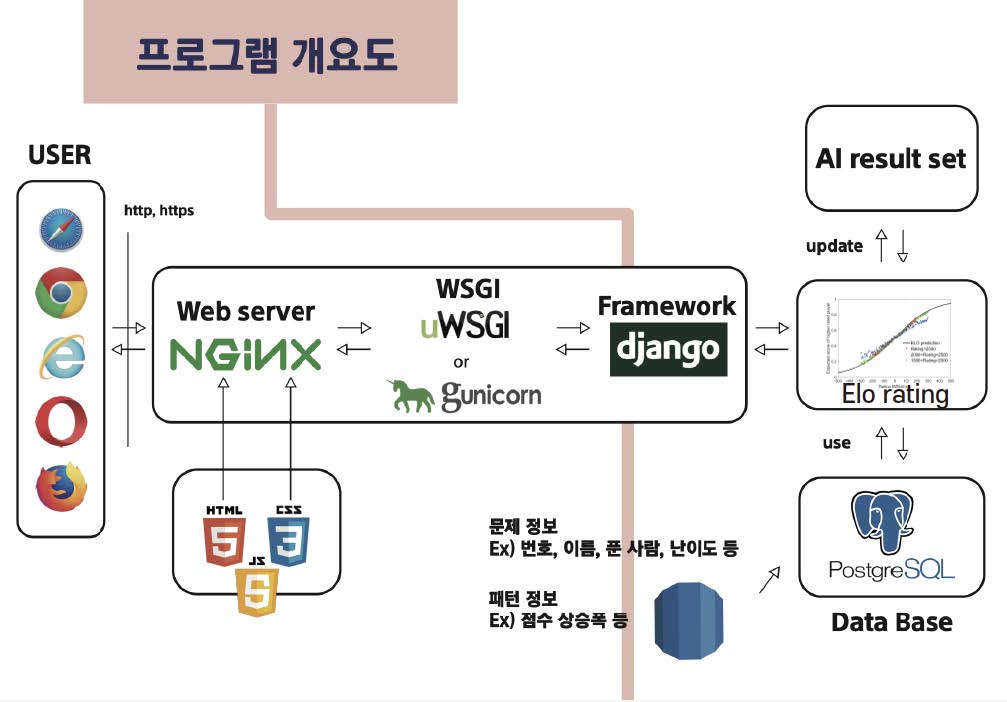
Server : AWS, NGINX, Gunicorn

DB : PostgreSQL

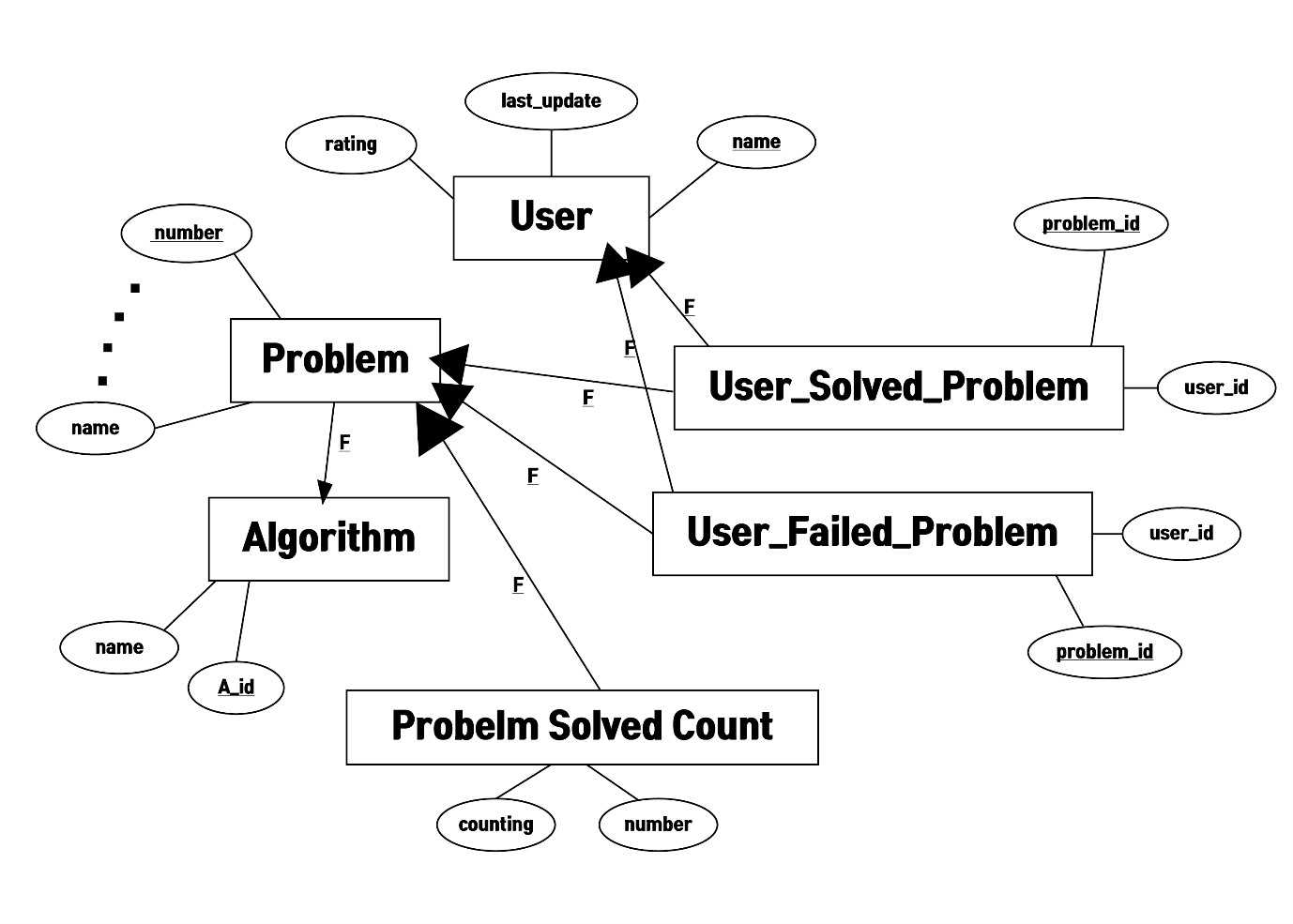
Language : Python3

Web : HTML, CSS, JavaScript

**2. 프로젝트 설명**

****

**2.1 데이터베이스**



**2.1.1 tb\_algorithm** - 백준 온라인 저지 사이트의 알고리즘 분류를 크롤링한 후 쓸모없는 것은 지우고 통합 가능한 분류는 합쳐서 알고리즘 분류별로 저장한 테이블

(algorithm\_id : 알고리즘 별 id, name : 알고리즘 분류 명)

**2.1.2 tb\_problem** - 백준 온라인 저지 사이트의 문제 데이터를 모두 크롤링하여 저장한 테이블

(number : 문제 번호, name : 문제 이름, submit : 제출, submit\_people : 제출한 사람, accept\_people : 맞은 사람, average\_attempt : 평균 시도, accept : 맞았습니다, wrong : 틀렸습니다, time\_over : 시간 초과, memory\_over : 메모리 초과, output\_over : 출력 초과, output\_type\_error : 출력 형식, runtime\_error : 런타임 에러, compile\_error : 컴파일 에러, accept\_proprotion : 정답 비율, accept\_people\_proportion : 정답률, algorithm\_id\_id : 알고리즘 분류)

**2.1.3 tb\_problem\_counting** – 각 문제별로 그 문제를 푼 유저들 중 몇 명이 다른 문제들을 풀었는지 카운트하여 저장한 테이블(problem\_id : 문제 번호, count\_array : problem\_id에 해당하는 문제를 푼 유저들이 1000번부터 18000번까지의 문제들 중 푼 문제들을 카운팅한 값을 저장해놓은 배열)

**2.1.4 tb\_user** – 약16만 명에 달하는 백준 온라인 저지 사이트의 이용자들의 데이터를 저장한 테이블

(name : 사용자 아이디, rating : 사용자의 점수, last\_update : 마지막 갱신 날짜 및 시간)

**2.1.5 tb\_user\_failed\_problem** – 유저들이 시도했지만 풀지 못한 문제 번호들을 저장한 테이블

(id : id, problem\_id : 문제 번호, user\_id : 사용자 아이디)

**2.1.6 tb\_user\_solved\_problem** – 유저들이 푼 문제 번호들을 저장한 테이블

(id : id, problem\_id : 문제 번호, user\_id : 사용자 아이디)

**2.2 서버**

사용자가 웹 브라우저를 사용해 서버에 접속하면 기본적으로 front-server인 nginx가 proxy server로서 활동합니다. 최전방에서 계산된 결과물을 보여주는 화면을 제외한 모든 동작은 back-end 단의 Django에서 모든 것을 연산하기 때문에 nginx가 load balancer의 역할을 해줍니다.

다만 nginx와 Django 단에서의 원활한 데이터 통신을 위해 wsgi 미들웨어가 필요한데 성능과 구현의 자유도는 uWSGI가 우수하지만 구현 난이도와 간편성, 그리고 소규모에서 큰 차이가 나지 않는 성능상의 이유로 gunicorn을 선택하였습니다.

Django 내에서 웹 템플릿은 Bootstrap 4를 기반으로 작성하였습니다. 따라서 다양한 디스플레이 크기에 대응할 수 있습니다.

**2.3 Elo rating system**

Elo rating system은 체스 등의 2명제 게임에서 실력 측정 및 평가 산출법입니다. 엘로는 이 산출법을 고안한 헝가리 태생 미국의 물리학자 아르파드 엘로에서 유래합니다. 체스계에서는 국제 체스 연맹의 공식 평점을 채용하는 등 강도를 나타내는 지표로 이용하고 있습니다.

이 수식을 우리가 필요한 방식으로 변형하여 Rating을 계산하는 식을 만들었습니다.

Rx = x유저의 현재 점수

R’x = x유저의 다음 점수

K = 점수 상승량 가중치

S = 풀이 여부

Ex = x유저가 풀 확률

**R’x = Rx + K(S - Ex)**

**2.4 문제 추천 시스템**

유저를 index화면에서 검색하면 검색된 유저가 푼 문제들의 번호를 데이터베이스에서 가져온다. 위에서 설명한 Problem\_counting 테이블에서 푼 문제들의 번호들의 count\_array를 가져와서 리스트에 추가한다. 각 인덱스에 있는 count\_array를 각 문제를 푼 사람의 수로 나누어서 문제를 풀 확률(승률)을 구한다.(예를 들어, 1000번 문제를 푼 사람의 수가 7000명이고 1000번을 푼 사람 중 1001번을 푼 사람의 수가 3500명이면 확률은 0.5가 된다. 이런식으로 모든 문제를 나눠준다.) 각 문제별로 count\_array를 나누어주었으면 각 문제의 count\_array의 인덱스(즉, 1000번 문제의 1000번 count값, 1001번 문제의 1000번 count값) 중 max값을 찾아서 result array에 하나씩 추가해준다. 여기서 max값을 찾는 이유는 어차피 자신이 푼 문제들 중의 count값을 보는 것이기 때문에 count값 중 낮은 값들은 고려할 필요가 없는 것이기 때문이다. max값들 찾아서 추가해주면 전체 문제의 수만큼 확률들이 있을 것이고 이 확률들 중 0.8에 가장 가까운 값을 찾아 문제를 추천해준다. 0.8이라는 수치는 어려운 문제와 쉬운 문제를 추천 해주기 보다는 어느정도 적정한 수준의 문제를 추천하는 것이 바람직하다는 가정하에 정한 것이다.

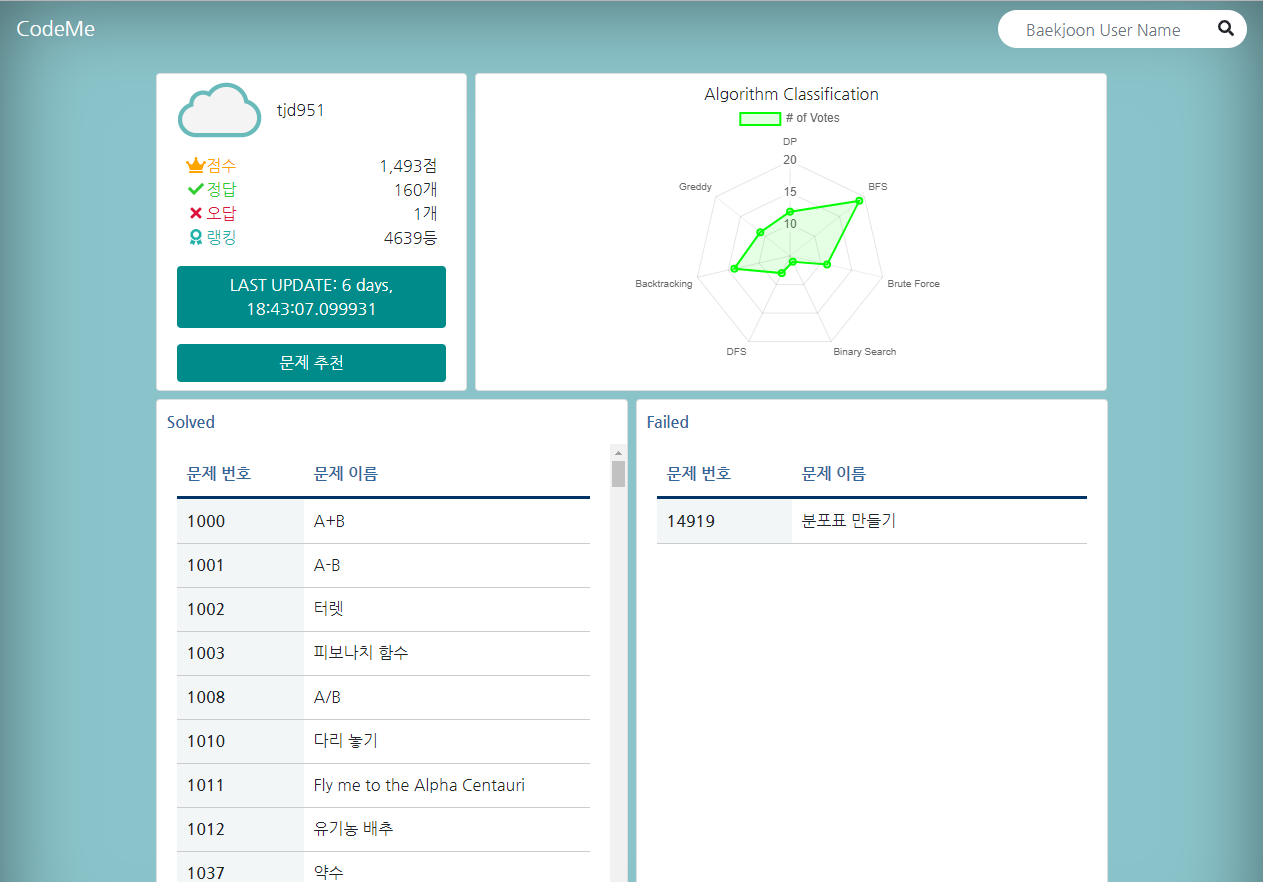
**2.5 웹페이지**

웹페이지는 ‘Atom’ 편집기를 이용하여 작성하였습니다. ‘HTML’, ‘CSS’, ‘JavaScript’를 모두 사용하였으며 ‘Bootstrap’ 프레임워크를 이용하여 개발하였습니다. 초기 웹페이지 개발 당시 여러 가지 편집기를 이용해 보고 싶어서 다수의 편집기(‘Bracket’, ‘Webstorm’)를 사용하였으나 최종적으로는 ‘Atom’으로 개발하게 되었습니다.

웹페이지는 크게 3가지 페이지로 나누어져 있습니다. 처음 사이트 접속 시 나타나는 유저 검색 화면, 다음으로 유저 아이디를 입력하면 나오는 유저 정보 페이지, 마지막으로 없는 유저 아이디가 입력될 경우 나오는 페이지로 구성되어 있습니다.

처음 접속 시 나오는 사이트는 Front사진과 유저 검색창 하단의 저작권 표시 배너로 구성되어 있습니다. Front사진에는 그간 개발해온 웹페이지 디자인 변화를 볼 수 있게 만들었고 사이트의 이름인 CodeMe를 반복적으로 넣었습니다. 유저 검색 화면에는 ‘백준 사이트’의 아이디를 입력하라는 메시지를 미리 나오게 만들었습니다.

두번째 페이지가 핵심 페이지입니다. 유저의 아이디가 입력되고 나서 유저의 ‘Rating 점수’, ‘푼 문제 개수’, ‘풀지 못한 문제 개수’, ‘등수’가 수치로 나타나고 아래의 ‘문제 추천’ 버튼을 누르게 되면 추천하는 문제가 있는 ‘백준 온라인 저지’ 사이트로 링크되어 이동합니다. 유저는 추천되는 문제를 풀고나면 새로운 문제를 추천받아 문제를 풀 수 있습니다. 하단에는 ‘Solved’와 ‘Not solved’로 나누어져 자기가 현재까지 풀고 못푼 문제들을 확인할 수 있습니다. 오른쪽 상단 영역에는 유저가 푼 문제를 알고리즘별로 분류하여 차트로 나타내었습니다. 어떤 알고리즘 분야의 문제를 적게 풀었는지 한눈에 볼 수 있게 하였습니다.



세번째는 없는 유저 아이디를 입력하였을 경우 존재하지 않는 사용자라는 경고 메시지와 그림이있는 페이지를 만들었습니다.



웹페이지 개발 담당이 하늘색을 좋아하여 전체적으로 하늘색 계통의 배경색을 입혔고 색감과 어울리는 구름, 고래 등의 그림을 확인할 수 있습니다.

**3. 프로젝트 강점**

**3.1 추천 시스템**

이 프로젝트가 기반으로 하고 있는 백준 온라인 저지 사이트에는 문제들이 업로드 된 순서대로 나열되어 있을 뿐 어떤 문제가 어렵고 쉬운지, 사람들이 어떤 문제를 더 선호하는지는 알기가 쉽지 않다. 이 프로젝트는 그런 어려움들을 해소할 수 있다. 풀지 않은 문제들 중에서 자신과 비슷하게 문제를 풀어나간 유저들의 데이터를 보고 문제를 추천해주기 때문에 일일이 문제를 찾지 않고도 유저들로부터 검증받은 문제를 추천받을 수 있다.

**3.2 성취욕 자극**

백준 온라인 저지 사이트에서는 단순히 푼 문제들의 개수로만 순위를 매기고 있다. CodeMe에서는 각 문제들의 개수뿐만 아니라 난이도 즉, 정답률을 보고 점수의 차등을 줬기 때문에 어려운 문제를 풀었을 때가 쉬운 문제를 풀었을 때보다 점수가 더 높다. 그래서 어려운 문제를 풀었을 때의 성취감을 보다 더 끌어낼 수 있다.

**3.3 손 쉽게 확인할 수 있는 정보**

백준 온라인 저지 사이트에서는 푼 문제들과 시도했지만 못 푼 문제들의 번호들을 나열만 했을 뿐 문제들의 제목은 보여주지 않는다. CodeMe에서는 이 부분을 손 쉽게 확인할 수 있다.

**3.4 깔끔한 UI**

눈에 자극적이지 않은 배경색과 간결한 선으로 풀어낸 카드박스, 다양한 그림으로 쉽게 알아 볼 수 있도록 깔끔하게 UI를 디자인하였다.

**4. 역할과 분담**

4.1 임진수 : 서버 관련 백엔드 작업

4.2 조재성 : 서버 구축 및 시스템 전반적인 작업

4.3 이성현 : DB 데이터 추가 및 관리

4.4 백상욱 : 웹 관련 프론트엔드 및 디자인 작업