

## 5조 Chromw Browser History 소스코드

### 1. DB 생성스크립트

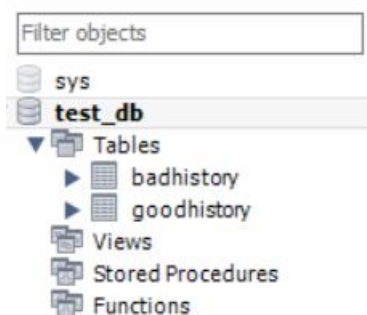
#### - Crawling\_csv.py

```
import csv
import pymysql

print("run crawling_csv.py")
def db_update(good_things:list, bad_things:list):
    try:
        with pymysql.connect(
            host="db",
            port=3306,
            user="devuser",
            password="devpass",
            charset="utf8",
            database="test_db"
        ) as connection:
            with connection.cursor() as cursor:
                create_goodhistory_table = """
                create table if not exists goodhistory(
                    id int auto_increment primary key,
                    contents varchar(500),
                    time varchar(100)
                );
                """
                create_badhistory_table = """
                create table if not exists badhistory(
                    id int auto_increment primary key,
                    contents varchar(500),
                    time varchar(100)
                );
                """
```

- DB(good,badhistory)테이블을 각각 생성

- DB테이블 생성화면



## 2. 크롤링 스크립트

### - history\_csv.py

```
import socket, sys, csv
import crawling_csv as cr

def socket_create():
    try:
        global host
        global port
        global s
        host = ''
        port = 9999
        s = socket.socket()
    except socket.error as err:
        print('socket create err : ' + str(err))

def socket_bind():
    try:
        global host
        global port
        global s
        print('bind socket to port : ' + str(port))
        s.bind((host,port))
        s.listen(5)
    except socket.error as err:
        print('socket create err : ' + str(err))
        # print('retry...')
        # socket_bind()

def socket_accept():
    conn, addr = s.accept()
    print('connection success')
    print('ip : ' + addr[0] + ' / port : ' + str(addr[1]))
    csv_transfer(conn)

    conn.close()
```

- socket create : 서버쪽 소켓을 생성합니다, 포트번호 9999
- socket bind : 주어진 포트번호와 호스트 주소를 바탕으로 해당 소켓에 포트번호를 바인딩 줍니다
- socket accpet : 클라이언트가 접속하게 되면 connection을 만들고 주어진 작업 수행  
(csv 파일을 클라이언트가 서버로 전송해주는 csv\_transfer 함수 호출)

- conn.close : 다 완료되면 연결 종료

```
def csv_transfer(conn):
    data = conn.recv(1024)
    if not data:
        print("no data")
        sys.exit()
    with open('web_history.csv', 'wb') as f:
        try:
            while data:
                f.write(data)
                data = conn.recv(1024)
        except Exception as e:
            print(e)
    print('csv file ready')

def history_process():
    good_things=[]
    bad_things=[]

    with open('web_history.csv','r', encoding='utf-8') as file:
        datas = csv.reader(file)

    # '클라우드' or '도커' or 'Splunk' or 'sql' or 'SQL' or 'Python' or '파이썬'
    for row in datas:
        if '클라우드' in row[0] or '도커' in row[0] or 'Splunk' in row[0] or 'sql' in row[0] or \
            'SQL' in row[0] or 'Python' in row[0] or '파이썬' in row[0] or 'splunk' in row[0] or 'vscod' in row[0]:
            good_things.append((row[0],row[1],))
        else:
            bad_things.append((row[0],row[1],))

    cr.db_update(good_things,bad_things)

    print('db update complete')
```

- csv\_transfer : 클라이언트가 연결된 후 서버쪽에서 클라이언트가 보내는 csv파일을 "recv" 함수를 통해 받아서 web\_history.csv 파일 생성.
- history\_process : 이걸 정민씨가 한거니깐 )) cr.db\_update 함수의 경우 crawling\_csv.py를 import 시킨 후 거기에 있는 db\_update 함수를 불러오는것  
(히스토리 프로세스 함수에서 처리한 굿핑즈와 배드핑즈 리스트를 crawling\_csv.py의 db\_update 함수에다 넘겨주고 db에 연동시키는 역할)

## - history\_client.py

```
import os, csv, datetime
import socket, sys
import subprocess
import sqlite3

def get_chrome_history():
    history_path = 'C:\\Users\\' + os.getlogin() + '\\AppData\\Local\\Google\\Chrome\\User Data\\Default\\History'
    con = sqlite3.connect(history_path)
    cursor = con.cursor()

    q = """
    SELECT title,datetime(last_visit_time/1000000-11644473600,'unixepoch','localtime') \
    FROM urls where last_visit_time >= ? order by last_visit_time asc
    """

    cur_time = (datetime.datetime(2021,
    datetime.datetime.today().month,
    datetime.datetime.today().day,
    9,
    0,
    0).timestamp()+11644473600)*1000000

    cursor.execute(q,(cur_time,))
    urls = cursor.fetchall()
    try:
        with open('web_history.csv', 'w', newline='', encoding='utf8') as hist:
            history_writer = csv.writer(hist)
            for line in urls:
                history_writer.writerow(line)
            print('chrome history parsing complete')
    except Exception as e:
        print('chrome history parsing error')
        print(e)
        sys.exit()
```

- get\_chrome\_history : 크롬 브라우저의 방문기록은 위에 주어진 history\_path 경로에 sqlite3 DB 파일로 저장되어 있습니다.
- os.getlogin : 현재 로컬PC 유저의 id를 가져오는 함수이며, path 경로를 통해 sqlite3 파이썬 내장 패키지를 이용하여 히스토리 DB를 불러오는 함수
- q : 히스토리 정보 읽어오는 sqlite3 select 쿼리문
- q의 쿼리문 : 크롬 히스토리는 webkit(웹킷) 시간으로 설정되어 있는데 (1601년 1월 1일 0시 0분 0초 기준 누적 시간을 마이크로세컨드로 표현), 이를 EPOCH 시간(에포크 시간 or 유닉스 시간) 으로 변환해주는게 저 숫자로 된 공식 (1970년 1월 1일 0시 0분 0초 기준 누적 시간을 초(seconds)로 표현), 에포크 시간으로 변환해야 datetime

함수로 우리가 쉽게 볼 수 있는 human date로 변환이 가능하기 때문입니다.

따라서 last\_visti\_time은 webkit 시간이니깐 이를 우선 뒤에 수식을 통해 EPOCH 시간으로 변환 후 datetime 함수에 넣어서 human date로 표현하도록 만들었고, 이것을 select 쿼리문으로 히스토리의 방문 사이트 이름(title)과 방문날짜(datetime 이하)를 수집하였습니다.

- cur\_time : 현재 시간을 webkit 시간으로 변경하는 용도. 크롬 히스토리에서 쿼리문 전송할때 현재 시간(현재 월, 일 오전 9시 기준) 이후로 쌓인 history를 긁어오기 위한것. q에 보시면 **where last\_visit\_time >= ?**로 된 부분이 있는데 거기에 cur\_time 값이 들어가서 그 시간 보다 큰(현재 날짜 오전 9시 이후) last\_visit\_time 값을 가진 데이터를 뽑아오게 합니다.

(webkit 시간으로 바뀌야 하는지? -> 크롬에서 webkit을 쓰기 때문에 history 긁어오는 쿼리문을 돌릴려면 webkit 시간 기준으로 해야합니다.)

- DB테이블에 데이터 삽입

The screenshot shows a database management interface with a left sidebar containing a tree view of database objects (sys, test\_db, Tables, badhistory, goodhistory, Views, Stored Procedures, Functions). The main area displays a SQL query: `select * from badhistory,goodhistory`. Below the query, a 'Result Grid' shows the results of the query. The grid has columns for id, contents, and time. The results are as follows:

id	contents	time	id	contents	time
1	1. PHP, mySql을 이용해서 홈페이지 만들기	2021-04-05 10:08:53	1	[pymysql] InternalError: 1366, "Incorrect String...	2021-04-05 09:17:08
1	1. PHP, mySql을 이용해서 홈페이지 만들기	2021-04-05 10:08:53	2	파이썬 MySQL Incorrect string value 에러 조치...	2021-04-05 09:22:19
1	1. PHP, mySql을 이용해서 홈페이지 만들기	2021-04-05 10:08:53	3	--default-authentication-plugin=mysql_native_p...	2021-04-05 09:26:46
1	1. PHP, mySql을 이용해서 홈페이지 만들기	2021-04-05 10:08:53	4	[Python][MySQL] Warnings: Incorrect string valu...	2021-04-05 09:37:16
1	1. PHP, mySql을 이용해서 홈페이지 만들기	2021-04-05 10:08:53	5	파이썬 크롤링 "Incorrect string value: 'WWWxED...	2021-04-05 09:37:27
1	1. PHP, mySql을 이용해서 홈페이지 만들기	2021-04-05 10:08:53	6	Python 에서 Mysql 에 한글, 특수문자 혹은 이모...	2021-04-05 09:37:38
1	1. PHP, mySql을 이용해서 홈페이지 만들기	2021-04-05 10:08:53	7	MySQL 8 on docker	2021-04-05 10:06:45
1	1. PHP, mySql을 이용해서 홈페이지 만들기	2021-04-05 10:08:53	8	mysql incorrect string value 특수문자 - Google ...	2021-04-05 10:07:38
1	1. PHP, mySql을 이용해서 홈페이지 만들기	2021-04-05 10:08:53	9	PHP #10_ 기본, MySQL 접속해서 데이터 가져오기	2021-04-05 10:15:35
1	1. PHP, mySql을 이용해서 홈페이지 만들기	2021-04-05 10:08:53	10	PHP로 DB값 불러오기 -1번방법 (mysql fetch a...	2021-04-05 10:15:50

```

def connection():
    try:
        s = socket.socket()

        host = 'localhost'
        port = 9999

        s.connect((host,port))
        try:
            with open('web_history.csv','rb') as f:
                data = f.read(1024)
                while data:
                    s.sendall(data)
                    data = f.read(1024)

            print('file transfer complete')
        except Exception as e:
            print('file transfer error')
            print(e)

    except Exception as e:
        s.close()
        print('Connection Close')
        sys.exit()

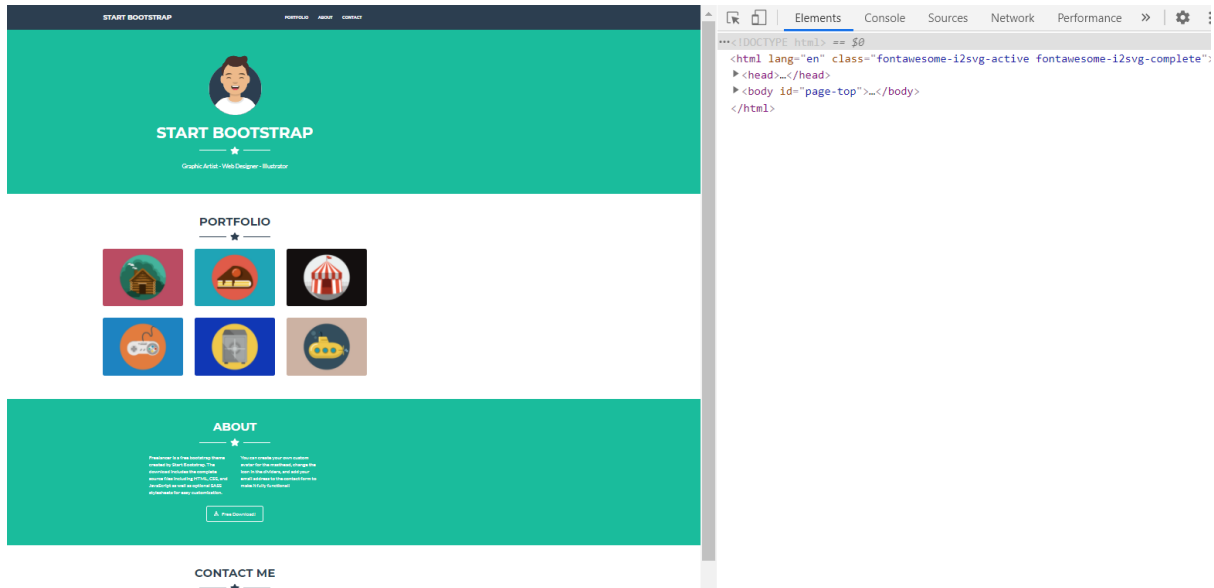
if __name__ == '__main__':
    get_chrome_history()
    connection()

```

- try ~ 구문 : web\_history.csv 파일 만들어서 쿼리로 불러온 히스토리 정보를 작성
- connection : 크롬 히스토리 파일을 csv로 다 만들고 나면 서버와 연결(s.connect), csv파일을 오픈해서 1024바이트씩 읽어온 후 sendall 함수를 통해 서버로 데이터 전송(바이트 스트림으로 전송)

### 3. 웹페이지 스크립트

- 무료 웹페이지 템플릿 다운로드



- 무료 웹페이지 템플릿 다운로드 후, html코드를 index.php에 그대로 가져와서 화면 설계한 내용대로 코드를 수정.

- bad.php

```
bad.php
<?php
$conn = mysqli_connect("db", "devuser", "devpass" , "test_db");
$sql = "SELECT contents, time FROM badhistory";
$result = mysqli_query($conn, $sql);
if (mysqli_num_rows($result) > 0) {
    while($row = mysqli_fetch_assoc($result)) {
        echo "비인가 웹히스토리: " . $row["contents"]. " 접속시간: " . $row["time"]. "<br>";
    }
}else{
    echo "테이블에 데이터가 없습니다.";
}
echo "Total: ", mysqli_num_rows($result);
mysqli_close($conn); // 디비 접속 닫기
```

- DB에 있는 goodhistory 테이블을 가져와서 모든 데이터를 출력해주는 파일

- good.php

```
good.php
<?php
$conn = mysqli_connect("db", "devuser", "devpass" , "test_db");
$sql = "SELECT contents, time FROM goodhistory";
$result = mysqli_query($conn, $sql);
if (mysqli_num_rows($result) > 0) {
    while($row = mysqli_fetch_assoc($result)) {
        echo "학습관련 웹히스토리: " . $row["contents"]. " 접속시간:" . $row["time"]. "<br>";
    }
} else {
    echo "테이블에 데이터가 없습니다.";
}
echo "Total: ", mysqli_num_rows($result);
mysqli_close($conn); // 디비 접속 닫기
?>
```

- DB에 있는 goodhistory 테이블을 가져와서 모든 데이터를 출력해주는 파일

- Index\_2.php

```
<?php
$conn = mysqli_connect("db", "devuser", "devpass" , "test_db");

$goodsql = "SELECT contents, time FROM goodhistory";
$badsql = "SELECT contents, time FROM badhistory";

$good_cnt = mysqli_num_rows(mysqli_query($conn, $goodsql));
$bad_cnt = mysqli_num_rows(mysqli_query($conn, $badsql));
$total_cnt = $good_cnt + $bad_cnt;

$result = (int)((($good_cnt / $total_cnt) * 100));

echo "전체학습률 : $result% <br>";
echo "학습태도 : ";
if($result >= 80){
    echo "우수";
}
elseif($result >= 60){
    echo "보통";
}
else{
    echo "나쁨";
}
echo "<br><br>";
echo "Total Good History Count : $good_cnt <br>";
echo "Total Bad History Count : $bad_cnt <br>";

mysqli_close($conn); // 디비 접속 닫기
?>
```

- DB에서 goodhistory, badhistory 데이터를 가져와서 전체학습률, 학습태도, 각 테이블 row count를 계산하여 echo를 통해 결과값들을 출력



- Index.php(웹페이지의 본체)

```
<!-- Portfolio Section Heading-->
<h2 class="page-section-heading text-center text-uppercase text-secondary mb-0">학습태도 분석결과</h2>
<!-- Icon Divider-->
<div class="divider-custom">
  <div class="divider-custom-line"></div>
  <div class="divider-custom-icon"><i class="fas fa-star"></i></div>
  <div class="divider-custom-line"></div>
</div>
<!-- 학습태도 분석결과-->
<div class="row justify-content-center">
  <? include ("index_2.php"); ?>
</div>
</div>
</section>
```

- <!--학습태도 분석결과--> 주석란에 index\_2.php를 삽입

(index\_2.php는 DB에서 goodhistory, badhistory 데이터를 가져와서 전체학습률, 학습 태도, 각 테이블 row count를 계산하여 출력해주는 파일)

```
<!-- 바로가기 페이지-->
<div class="text-center mt-4">
  <a class="btn btn-xl btn-outline-light" href="good.php">
    | 학습 히스토리
  </a>
  <a class="btn btn-xl btn-outline-light" href="bad.php">
    | 비인가 히스토리
  </a>
</div>
</div>
```

- 각각 goodhistory, badhistory의 데이터들을 모두 출력해주는 페이지로 각각 이동하기 위한 버튼을 생성

## - Docker-compose.yml

```
version: '3.7'
services:
  web:
    build: .
    container_name: php73
    depends_on:
      - crawling
    volumes:
      - ./var/www/html/
    ports:
      - 8282:80
  db:
    container_name: mysql5
    hostname: db
    image: mysql:5.7
    command: --default-authentication-plugin=mysql_native_password --character-set-server=utf8mb4 --collation-server=utf8mb4_unicode_ci
    environment:
      MYSQL_ROOT_PASSWORD: root
      MYSQL_DATABASE: test_db
      MYSQL_USER: devuser
      MYSQL_PASSWORD: devpass
    ports:
      - 3306:3306
  crawling:
    build: ./crawling
    hostname: crawling
    depends_on:
      - db
    environment:
      crawling_DB_HOST: db:3306
    ports:
      - 9999:9999
    links:
      - db
```

## - Web 서버 구축 도커파일

```
php > Dockerfile > ...
1 FROM php:7.3.3-apache
2 RUN apt-get update && apt-get upgrade -y
3 RUN docker-php-ext-install mysqli
4 EXPOSE 80
```

## - 크롤링 서버 구축 도커파일

```
php > crawling > Dockerfile > ...
1 FROM python
2 WORKDIR /code
3 RUN apt-get update -y && apt-get upgrade -y
4 RUN pip install pymysql
5 COPY . /code/
6 RUN chmod 777 /code/run.sh
7 CMD /code/run.sh
```

## - 실행 순서

1. 현재 작업 디렉토리에서 history\_client.py 실행
2. Docker-compose up 실행
3. Localhost:8282/index.php 접속 후 데이터 확인

The screenshot displays the 'CROME BROWSER HISTORY' application interface. At the top, there's a header with a user profile icon and the title 'CROME BROWSER HISTORY'. Below this, a section titled '학습태도 분석결과' (Learning Attitude Analysis Results) shows a progress bar at 20% and statistics: 'Total Good History Count: 44' and 'Total Bad History Count: 103'. A '상세결과' (Detailed Results) section has two tabs: '학습 히스토리' (Learning History) and '비인가 히스토리' (Unauthorized History). Below these tabs, two browser windows are shown side-by-side. The left window, titled 'localhost:8282/good.php', displays a list of browsing history items categorized as '학습관련' (Learning-related). The right window, titled 'localhost:8282/bad.php', displays a list of browsing history items categorized as '비인가' (Unauthorized).

**학습관련 히스토리:** 클라우드 보안 융합 전문가양성과정 - Google Docs 접속시간:2021-04-05 09:37:42  
학습관련 히스토리: 클라우드 보안 융합 전문가양성과정 - Google Docs 접속시간:2021-04-05 10:09:08  
학습관련 히스토리: Docker compose가 Windows에서 실행되는 데 너무 오래 걸리는 이유는 무엇입니까? 접속시간:2K  
학습관련 히스토리: docker compose 실행이 너무 오래 걸리는 이유는 무엇입니까? 접속시간:2021-04-05 10:24:12  
학습관련 히스토리: Docker compose가 Windows에서 실행되는 데 너무 오래 걸리는 이유는 무엇입니까? - Javaer101  
학습관련 히스토리: 클라우드 보안 융합 전문가양성과정 - Google Docs 접속시간:2021-04-05 13:39:36  
학습관련 히스토리: MySQL 한글 insert시 발생하는 ERROR 1366 (HY000): Incorrect string value: 문제 :: 코드 조각-An  
학습관련 히스토리: MySQL 테이블에 데이터 한글 입력 시 오류 해결 접속시간:2021-04-05 16:09:25  
학습관련 히스토리: [DB/MariaDB/MySQL] MySQL 언어설정으로 인한 문제, 그리고 해결 접속시간:2021-04-05 16:10:56  
학습관련 히스토리: docker mysql default-character-set - Google 검색 접속시간:2021-04-05 16:29:51  
학습관련 히스토리: docker mysql 한글 깨짐 해결 & 초기 설정 | epr 접속시간:2021-04-05 16:29:55  
학습관련 히스토리: mysql dockerfile character set - Google 검색 접속시간:2021-04-05 16:32:18

**비인가 히스토리:** Launch Meeting - Zoom 접속시간:2021-04-05 09:02:05  
비인가 히스토리: Launch Meeting - Zoom 접속시간:2021-04-05 09:02:05  
비인가 히스토리: 이후 참석자 - Zoom 접속시간:2021-04-05 09:07:06  
비인가 히스토리: 이후 참석자 - Zoom 접속시간:2021-04-05 09:07:06  
비인가 히스토리: 반디집 - Google 검색 접속시간:2021-04-05 09:14:31  
비인가 히스토리: 기록/순위, 농구: 네이버 스포츠 접속시간:2021-04-05 11:40:48  
비인가 히스토리: nba: 네이버 통합검색 접속시간:2021-04-05 14:11:15  
비인가 히스토리: 기록/순위, 농구: 네이버 스포츠 접속시간:2021-04-05 14:11:28  
비인가 히스토리: 기록/순위, 농구: 네이버 스포츠 접속시간:2021-04-05 14:11:28  
비인가 히스토리: 기록/순위, 농구: 네이버 스포츠 접속시간:2021-04-05 14:11:30  
비인가 히스토리: 받은메일함(8718): 네이버 메일 접속시간:2021-04-05 14:12:37