REPORT

성결대학교
SUNGKYUL UNIVERSITY

Profiler
구현 보고서
(수 4-6 교시)

자바웹 A++++ 팀

컴퓨터공학과 20180876 천민우
컴퓨터공학과 20190274 정택원
컴퓨터공학과 20210833 남민지
컴퓨터공학과 20210839 박선하

<목차>

<그림, 표 목차>

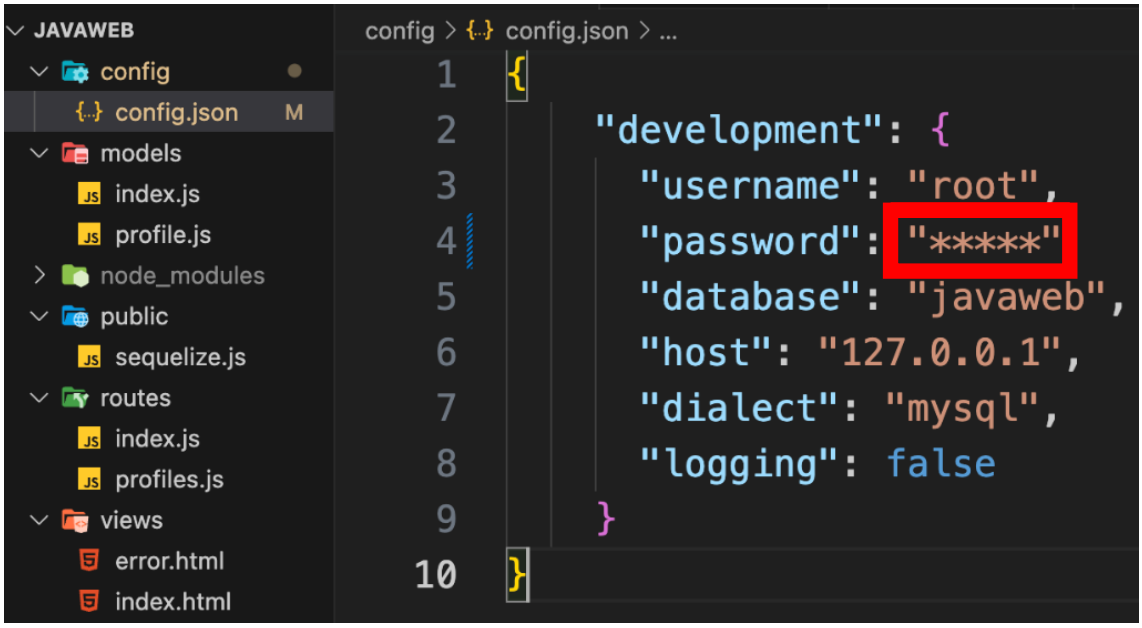
1. 프로그램 개요

컴퓨터의 무거운 작업은 CPU Core의 개수에 의존한다. Core의 개수는 물리적으로 한정되어 있어서 Core의 Task처리 속도가 중요하다. 이를 위해, 엔지니어들은 빈틈 없는 작업을 수행하도록 하는 Core를 개발한다. 여러개의 코어가 수행한 task의 양(결과물)을 한 눈에 볼 수 있도록
프로파일링을 하여 시각화한다.

본 팀은 SW개발에서 사용되는 CPU Core의 성능 프로파일러를 이번 자바 웹 프로그래밍 수업에서 배운 Node.js를 활용해 구현하였다. 프로젝트의 동작 과정과 수집한 데이터를 처리하기위해 Parsing, 정제된 데이터를 Database에 저장, DB에 저장된 데이터를 호출하여 각 Core에서 얼마만큼의 수행능력을 보이는지 시각화하는 과정을 직접 살펴보며 Node.js에서 서버 I/O의 작동방식을 살펴본다.

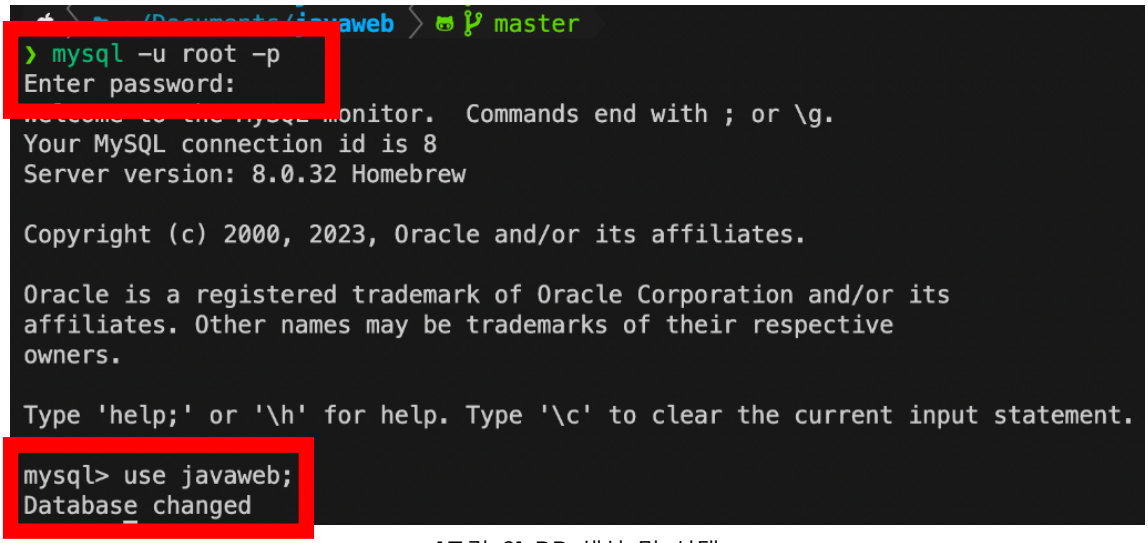
2. 프로그램 수행 절차

2.1 사용 방법 안내



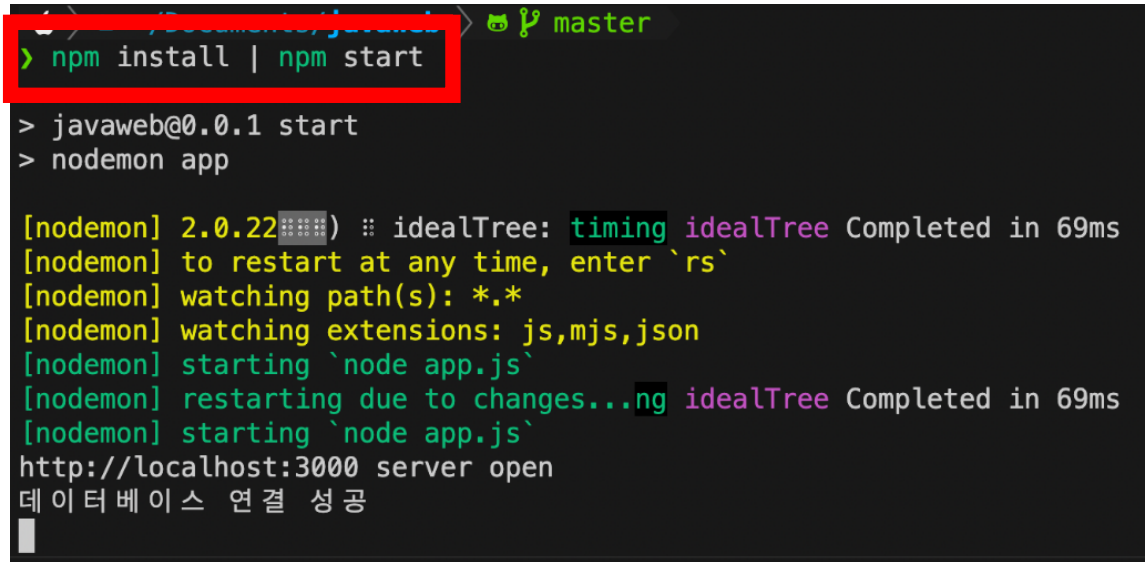
[그림 1] mysql 비밀번호 수정

프로젝트의 javaweb/config/config.json 파일로 이동하여 현재 입력되어 있는 password를 컴퓨터에 설치된 mysql의 비밀번호로 변경한다.



[그림 2] DB 생성 및 선택

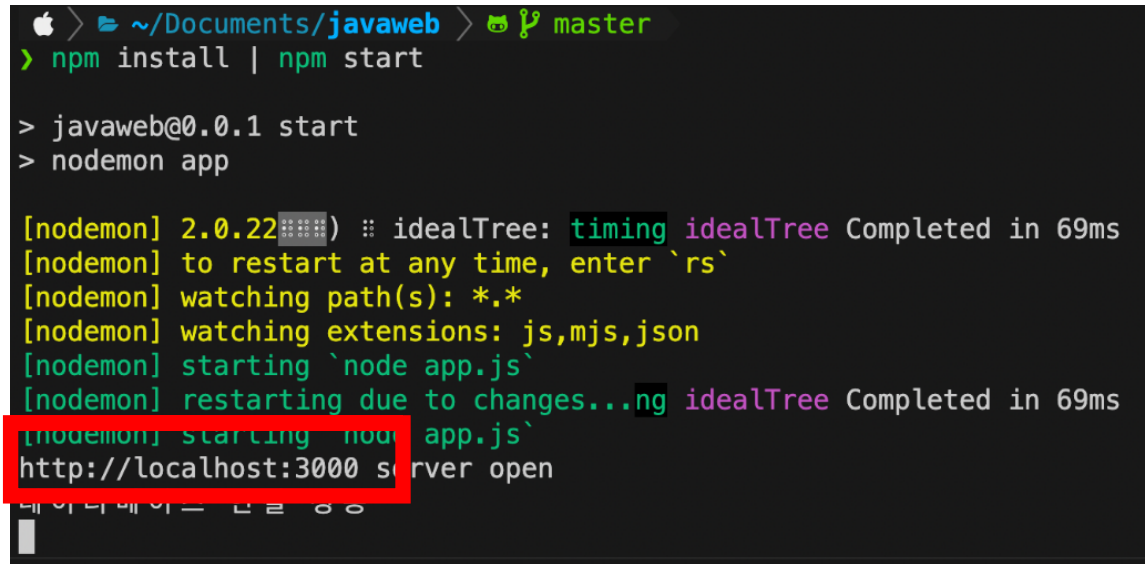
터미널에서 root계정으로 mysql에 접속한 후, use javaweb; 명령어를 통해 javaweb 데이터베이스를 생성하고 사용한다. 위 그림과 같이 콘솔의 출력이 확인되면 exit 명령어를 통해 mysql을 종료한다.



[그림 3] npm 설치 및 프로젝트 실행

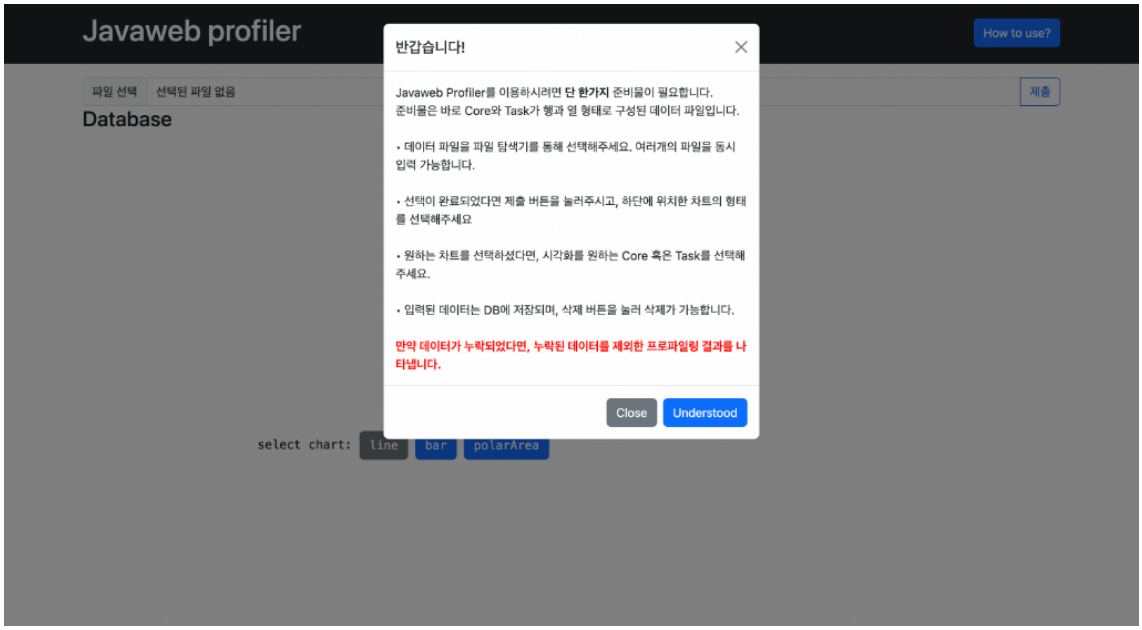
node_modules 폴더에 있는 패키지를 다운받기 위해 npm install 명령어를, 프로젝트를 바로 실행시키기위해 npm start 명령어를 |(파이프)를 이용해 동시에 명령한다.

2.2 프로그램 시작



[그림 4] URL 접속

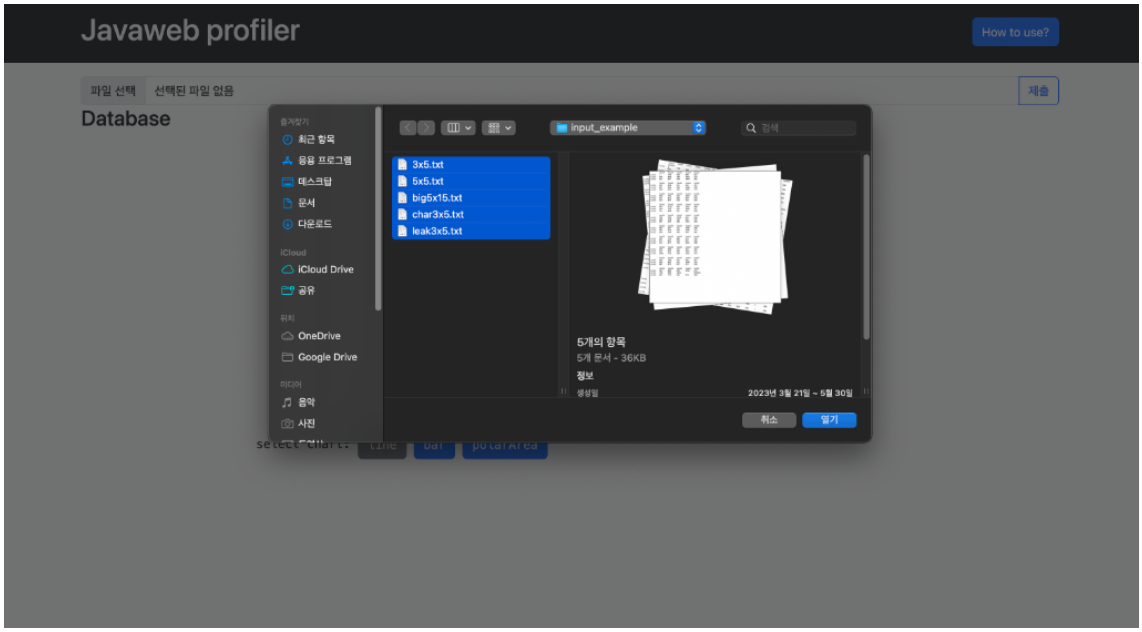
터미널에 출력되는 주소에 마우스 커서를 가져가서 Crtl + Click (mac: Command + Click)을 하여 프로젝트 페이지로 접속한다.



[그림 5] Profiler 사용 설명서

화면의 우상단 How to use? 버튼을 클릭하여 프로젝트의 간단 사용 방법을 숙지한다. 확인 후 Close 버튼을 클릭하여 설명서를 닫는다.

2.3 데이터 입력



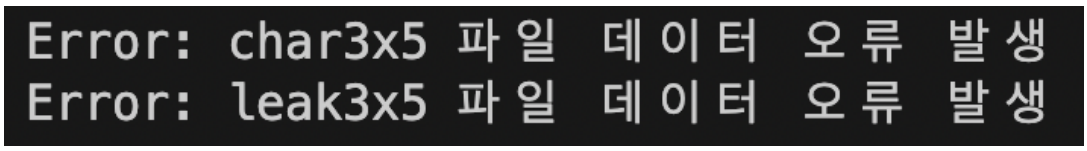
[그림 6] 파일 선택 (다중 선택 화면)

파일 선택 버튼을 클릭하여 Profiling을 원하는 데이터 파일을 선택한 후 제출 버튼을 클릭한다. 단일 파일 입력뿐만 아니라, 다중 파일 입력 또한 가능하다.

[그림 7] 데이터 입력 결과에 따른 alert 메시지

입력된 파일에 따라 alert를 통해 메시지를 출력한다. 이미 존재하는 이름의 데이터 파일을 제출하게 되면, DB의 무결성을 위해 제출할 수 없다.

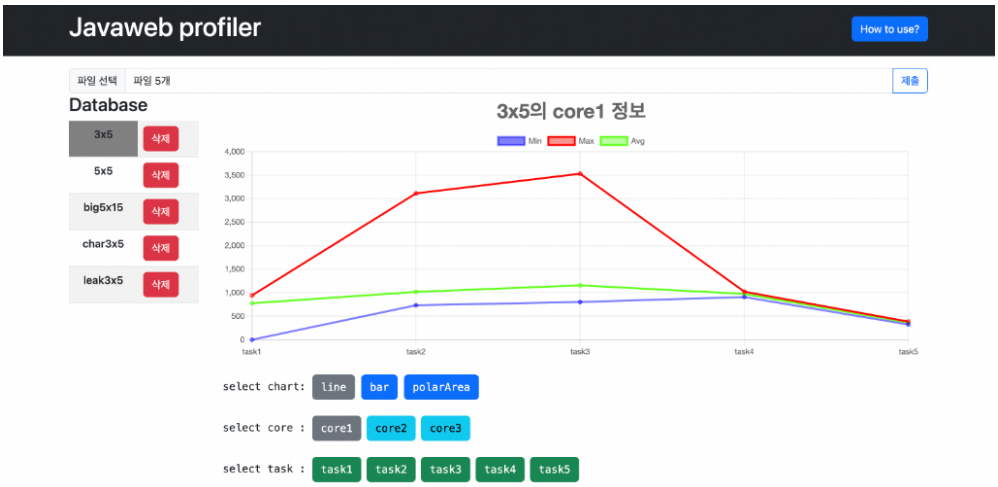
2.4 오류 처리



[그림 8] 데이터 파일 오류 발생시 서버 콘솔 출력 메시지

만약 데이터 파일에 숫자가 아닌 공백, 문자가 입력된다면 오류를 처리해야 한다. 본 팀은 해당 오류가 발생하면 서버의 콘솔에 오류 메시지를 출력하며, 오류가 발생한 행x열 만 DB에 삽입되지 않고 나머지 행x열은 정상적으로 저장될 수 있도록 처리하였다.

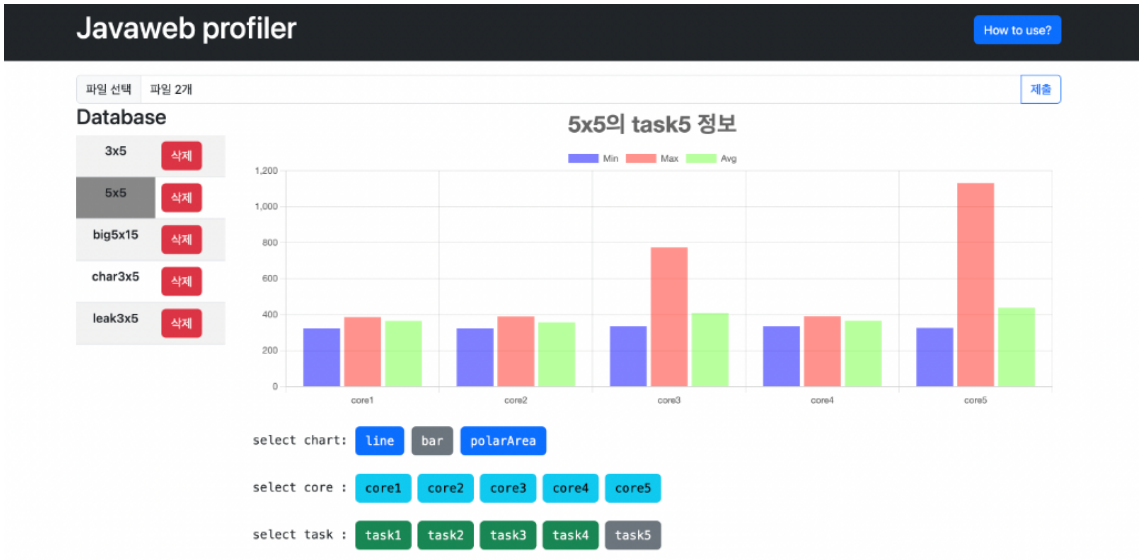
2.5 결과 출력



[그림 9] 3x5 데이터 프로파일링 결과; line 형태 차트

파일이 정상적으로 저장되면, 좌측에 입력된 데이터들의 리스트를 선택할 수 있다. 데이터 선택 후 원하는 차트 형태를 선택하고 core와 task를 클릭하면 위와 같은 프로파일링 결과 차트를 확인할 수 있다.

2.6 버튼 동적 생성



[그림 10] 5x5 데이터 프로파일링 결과; bar 형태 차트

[그림 9] 에서와 달리 버튼의 수가 5개씩 생성된 것을 볼 수 있다. 버튼은 DB의 core, task 개수에 따라 동적으로 생성된다. 출력을 원하는 차트를 선택 후 core 또는 task 번호를 클릭하면 각 데이터에 대한 프로파일링 한 최소, 최대, 평균값 결과 정보를 세가지 차트로 제공한다.

2.7 데이터 삭제

[그림 11] 삭제 버튼을 통한 데이터 삭제

데이터베이스가 나열되어있는 리스트의 삭제 버튼을 클릭하면, 데이터베이스를 삭제할 수 있다.

3. 프로그램 기능

3.1 Node 서버 구성

노드 프로젝트의 루트 폴더에 app.js를 통해 서버를 실행한다. app.js는 express, morgan, path, nunjucks, sequelize 외부 라이브러리를 사용하며, 그 중 express 라이브러리를 통해 서버를 구동한다. morgan 라이브러리는 원활한 개발을 위해 접속 로그를 확인하는 용도로 사용하였으며, 현재는 주석처리되어 비활성화된 상태이다.
서버에는 config, models, node_modules, public, routes, views 폴더가 존재한다. 각 폴더의 역할은 다음과 같다.

[표 1] 구성 폴더 및 폴더 역할

폴더명	역할
config	데이터베이스와 연결하기 위한 정보가 포함된 config.json파일 저장공간
models	MySQL의 데이터를 노드서버에서 사용하기 위한 sequelize파일들 저장공간
node_moduels	npm을 통해 다운받은 외부라이브러리 저장공간
public	프론트엔드에서 사용하는 파일 저장공간 static으로 지정
routes	express.Router를 사용한 라우터 파일들 저장공간
views	프론트엔드 HTML 코드 저장공간

세부 주소 없이 접속된 사용자는 routes폴더 내부의 index.js 라우터로 연결된다. index.js 라우터는 DB로부터 테이블 목록을 조회한 데이터를 포함하여 views/index.html을 render한다.

3.2 데이터 파일 입력

index.html파일은 static으로 지정된 public폴더에서 sequelize.js파일을 불러와 사용한다. 서버에서 전달된 테이블리스트를 화면에 출력하는 nunjucks 반복문을 포함한다.
프론트엔드 페이지에는 여러개의 파일을 한번에 입력할 수 있는 <input>태그가 존재한다. 제출 버튼에는 이벤트 리스너가 연결되어 있다. 만약 선택 파일 없이 제출 시도시 파일이 없다는 alert가 호출된다.
입력된 파일은 서버로 바로 전송되는 것이 아닌 프론트엔드 자바스크립트 코드를 거친다. 각각의 파일을 검사하여 해당 파일의 확장자가 txt인지 확인한다. txt형이 아닌 다른 파일이 포함되어 있으면 alert를 통해 경고하며 서버로 데이터가 전송되지 않는다. txt파일이라면 바로 서버로 전송되는 것이 아닌, 기본적인 파싱 과정을 거치기 위해 readTextFile이 호출된다.
text파일을 줄바꿈(\n) 기준으로 split하고, split된 인덱스 각각에 양쪽 공백을 제거 후 탭(\t), 띄어쓰기, 콤마(,), 슬래시(/)를 기준으로 다시 split한다. 각각의 배열은 파일의 이름을 첫번째 인덱스로 갖는 배열에 다시 삽입된다. 이 과정을 통해 생성된 2차원 배열을 다시 배열에 삽입해 각 파일의 정보가 들어가 있는 3차원 배열이 생성된다. 배열은 /profiles 주소로 post

메소드를 통해 비동기로 전송된다.
 해당 요청은 routes/profiles.js 라우터에서 처리한다. 파일의 이름을 각각 체크하여 이미 데이터베이스에 동일한 이름을 가진 테이블이 있다면 저장하지 않는다. models/index.js 에 선언된 createDynamicTable을 호출하여 새 테이블을 생성한다. 각각의 프로파일 데이터와 core정보, task정보를 한 행으로 하여 새로 생성된 테이블에 삽입한다. 만약 누락되었거나 잘못된 형식의 데이터가 있다면 해당 데이터는 무시하고 수행한다. 생성된 테이블의 개수 정보를 담아 호출되었던 프론트엔드 코드로 반환된다.
 프론트엔드에서는 반환된 메세지를 alert로 출력하여 사용자에게 저장 정보를 알린 후 테이블 리스트의 출력을 갱신한다.

3.3 차트 출력 과정

1. 테이블 리스트의 정보를 선택하면 서버에 core와 task의 정보를 요청한다.

2. 서버는 연결된 DB에서 선택된 테이블의 core정보와 task정보를 각각 확인한다. DISTINCT로 쿼리를 전송하여 각각의 core와 task 정보를 중복 없이 받는다.

3. 전송받은 정보로 해당 프로파일의 core와 task 버튼을 동적으로 생성한다.

4. 생성된 core 또는 task의 버튼을 선택시 파일 이름과 선택한 버튼의 정보를 바탕으로 서버에 그래프를 만들기 위한 데이터를 요청한다.

5. 서버는 요청된 데이터를 조건으로 group별 min, max, avg 데이터를 요청하는 쿼리문을 실행한다.

6. 데이터베이스 검색이 완료되면 해당 데이터를 json형태로 프론트엔드에 반환한다.

7. 프론트엔드는 반환받은 데이터를 바탕으로 chart를 화면에 그린다.

4. 프론트엔드 수행 절차 및 로직

4.1 버튼 동적 생성

데이터베이스를 선택하면 해당 DB의 core, task 수에 따라 프론트엔드에서 보이는 버튼의 수가 달라진다. javaweb/public/sequelize.js 파일의 155line getdata()함수를 통해 버튼의 동적 생성이 가능해진다. map() 함수를 통해 수에 따라 <button>태그를 반복적으로 생성하며, 태그의 class는 bootstrap5의 'btn' 클래스를 부여하여 시각적으로 다채로운 색의 버튼을 구현한다.

4.2 chart.js

차트 시각화는 <script>태그 내부에서 CDN을 이용하여 chart.js 라이브러리를 사용하였다. 여러 라이브러리의 docs를 읽어보니, 사용자로서 읽기 편했고 사용방법이 간편하였기 때문에 해당 라이브러리를 사용하였다.

getdata() 함수가 실행되면 서버로부터 원하는 데이터를 배열 형태로 반환받아, chart.js에 맞게 입력을 해주면 자동으로 시각화를 진행한다.

4.3 bootstrap5

프론트엔드의 정돈된 인터페이스를 위해 <script>태그 내부에서 CDN을 이용하여 bootstrap5 라이브러리를 사용하였다. docs를 읽어보면 원하는 모양별로 class가 정의되어있어, 검색을 통해 원하는 인터페이스를 구현할 수 있었다.

본 팀의 Profiler를 사용하기 위해서는 사용 설명서가 필요하였다. bootstrap5의 staticBackdrop modal을 이용해, 사용자가 사용법을 정확히 숙지할 수 있도록 가이드를 제공하였다.

5. about version

5.1 ver1.0.0 동작 방식

과제의 주제가 주어졌을때 수업 내용 바탕이 아닌, 인터넷 서칭을 통해 제작한 version이 있다.

해당 버전은 mongoDB 클러스터를 사용하여 nodejs 데이터베이스를 생성한다. 서버 내부에서 core 1 ~ core N에 대한 컬렉션과 task 1 ~ core N에 대한 컬렉션을 생성한다. 컬렉션 내부에 생성 될 데이터들은 name(string)과 ary(array)를 속성으로 구성되어있다.

core에는 (task의 수) x (데이터의 수) 만큼의 int형 데이터가 삽입된다. 데이터를 호출할 때에는 0번째 인덱스에 빈 배열을 넣고 task의 수로 모듈러 연산(%)을 하여 데이터를 반환하는 형태로 구현하였다.

더 자세한 로직은 **깃허브**[1]에서 확인 가능하다. README를 읽고 가이드에 따라 실행하면, 정상적으로 동작이 가능하다.

5.2 ver2.0.0

모든 수업을 수강하고 교수님의 7장 동영상과 교재에 나온 데이터베이스인 Mysql은 mongoDB와 어떻게 다른가 확인하고싶어 Mysql을 써서 프로젝트를 develop하였다. 입력된 데이터의 작업량 정보를 한번씩만 저장해야 무결성이 유지될 수 있다고 생각하였다. Mysql을 사용하여 각각의 core, task에 대한 row를 생성한다.

ver1.0.0에서는 단일 파일 처리만 가능했다. 개선된 버전에서는 다중 파일 처리가 가능하다. 여러 파일을 한번에 제출하면, 파일의 개수에 맞추어 데이터베이스를 생성한다. 생성된 데이터베이스는 리스트형태로 인터페이스에서 제공되며, 여러 프로파일링 결과에 쉽게 접근 가능하다. 또한 데이터베이스 삭제에 대한 권한도, 인터페이스에서 제공된다.

프론트엔드 부분에서의 개선점은 버튼을 클릭했을때 버튼의 색이 변환되는 부분이 있다.

1) https://github.com/javawebApppp/javaweb_1.0.0 (ver1.0.0)

6. 프로젝트 기여

[표 2] 자바웹애플리케이션 Profiler구현 과제 기여

20210833 남민지(25)	20190274 정택원(25)	20180876 천민우(25)	20210839 박선하(25)