**[6조: 진실한 남매들]**

**프로젝트 결과 보고서**

프로젝트명: <진실씨의 진실한 간식추천>

**1. 팀 소개**

진실한 남매들은 팀장님 김진실 씨와 배달의 민족을 만든 회사인 우아한 형제들을 합쳐 만든 이름입니다. 팀명처럼 사용자들에게 진실만을 전해주는 프로그램을 만들기 위해 노력하였습니다.

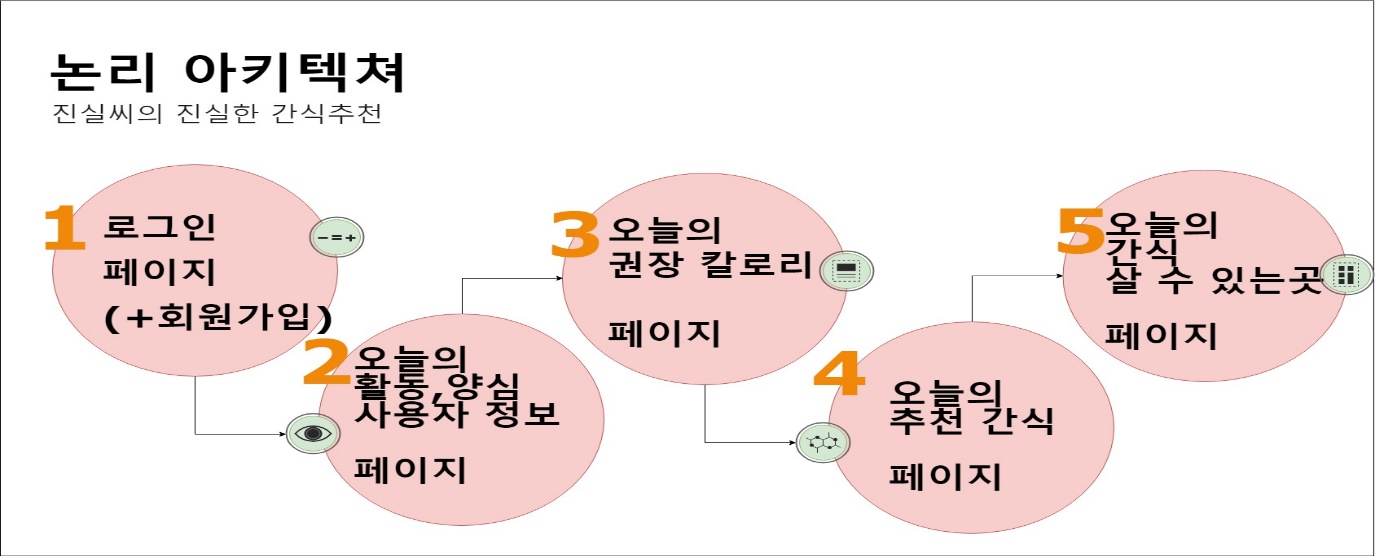
|  |  |
| --- | --- |
| 김진실© | JAVA프로그래밍, 서버구축 조사 |
| 김지효 | JAVA프로그래밍, 데이터조사 |
| 김태리 | API연동 |
| 이태희 | 서버구축, 시연 |
| 정민구 | 데이터모델링, 데이터구축, 발표 |

**2. 프로젝트 주제**

진실씨의 진실한 간식추천은 사용자의 정보를 받아 섭취할 수 있는 칼로리를 계산하여 그에 맞는 간식을 추천해주는 웹 페이지입니다.

**3. 프로젝트 개요**

요약 개요:

****

세부 설명:

1. “진실씨의 진실한 간식추천” 로그인 페이지로부터 시작.
   1. 아이디/비밀번호가 없다면 회원가입 페이지로 넘어갑니다.
      1. 아이디/비밀번호 및 개인정보는 서버의 DBMS에 저장되며 로그인 시 활용.
   2. 회원가입 완료 시, 입력된 회원정보 출력 창 띄워 짐 (로그인 페이지 진행 가능)
2. 로그인 성공 시, 환영 페이지 출력
   1. [오늘의 칼로리는?] 버튼 클릭
3. “오늘의 활동, 양심 입력 페이지”에서 사용자의 정보, 활동량 및 양심정도 선택.
   1. [오늘의 권장 칼로리는?] 버튼 클릭
4. “오늘의 권장 칼로리 페이지” 에서 입력값을 계산, 권장 칼로리를 출력합니다.
   1. [오늘의 간식 추천받기] 버튼 클릭
5. “오늘의 추천 간식 페이지” 에서 권장 칼로리 확인 및 DBMS에서 칼로리에 맞는 간식 3개를 추천
   1. [살수 있는 곳] 버튼 클릭
6. “오늘의 간식 위치 페이지” 간식 위치 지도 출력.

**4. 활용범위**

단순한 칼로리 계산이 아닌 활동량과 양심정도에 근거하여 사용자에 맞춘 간식을 추천한다.

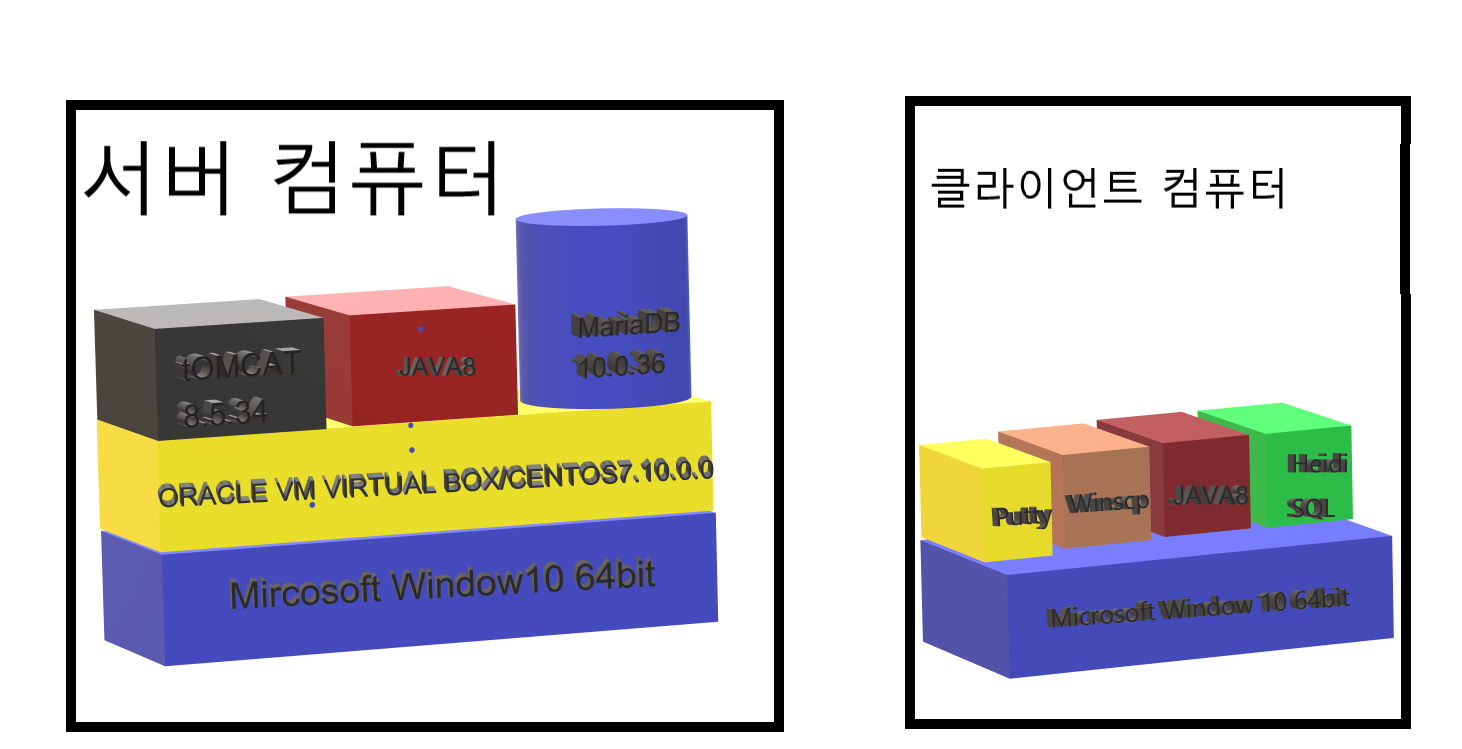
이로써 사용자도 정확히 계량화 할 수 없는 자신의 데이터를 진실한 남매들의 개발자들이 독창적으로 고안한 비밀공식을 이용하여 풀어내 더욱 정확한 결과물을 도출 할 수 있다.

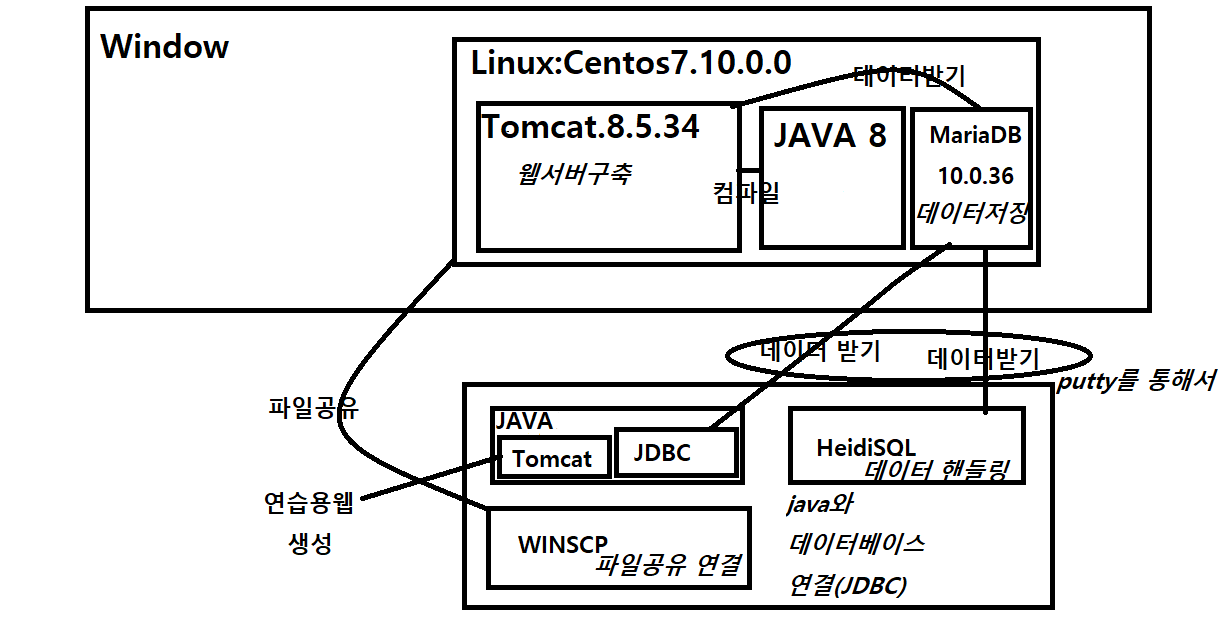
**5. 개발내용**

**List:**

* **서버구축 Part**
* **JAVA 개발 Part**
* **API 개발 Part**
* **DB 모델링 Part**

**서버구축 Part:**

****



<Tomcat: Tomcat 8.5.34 사용>

Tomcat 사용 배경:

* 자바로 만든 파일을 웹 서버에 띄우기 위해서는 자바파일을 읽을 수 있는 웹서버가 필요.

(아파치는 html만을 읽을 수 있으므로 jar파일을 열어 보기 위해서는 tomcat이 필요)

Tomcat 세팅 및 실행 방법:

* wget명령어를 이용하여 톰캣 tar파일(압축파일) 설치
* tar명령어를 이용하여 톰캣 압축해제
* -A INPUT -m state --state NEW -m tcp -p tcp --dport 8080 -j ACCEPT 을 이용하 8080 열기
  + (톰캣은 8080포트를 사용)
* ./startup.sh 를 이용하여 톰캣을 실행해준다.
* [http://localhost:8080](http://localhost:8080/)으로 접속

<JRE: JAVA8 사용>

JAVA 8 사용 배경:

* jar파일을 컴파일 하기 위해서 JAVA8을 설치

JAVA 8 설치 및 세팅:

* wget명령어를 이용하여 java 8버전 설치한다.
* tar명령어를 이용하여 압축해제한다.
* VI 편집기 이용classpath에 JAVA를 추가한다.

<MariaDB: MariaDB10.0.36사용>

MariaDB 사용배경:

* 데이터를 다룰 시, 데이터를 담을 수 있는 통이 필요. 이 역할을 하는 것이 MariaDB임.

MariaDB 설치 및 설정:

* yum repolist를 이용하여 MariaDB repolist가 유무 여부 확인.
  + //etc/yum.repos.d/MariaDB.repo를 편집기를 이용하여 오픈.
  + (버전에 맞게 추가)
* MariaDB repolist 설정값 입력으로 저장소 추가.

[mariadb]

name = MariaDB

baseurl = http://yum.mariadb.org/10.0/centos7-amd64

gpgkey=https://yum.mariadb.org/RPM-GPG-KEY-MariaDB

gpgcheck=1

* 저장소 설정값 입력 후, MariaDB를 설치.
  + yum install MariaDB-server MariaDB-client

MariaDB 시작:

* 마리아 DB 시작service mysql start로 MariaDB를 시작.

MariaDB 사용자 접속 권한 부여:

* 루트계정의 비밀번호를 설정
  + /usr/bin/mysqladmin -u root password '비번입력'
  + mysql -uroot -p를 입력 후, 비밀번호를 입력 시, MariaDB에 root계정으로 접속 가능.
* 외부 접속 허용 설정 및 서버 컴퓨터 IP
  + GRANT ALL PRIVILEGES ON \*.\* TO root@'자신의 아이피IP' IDENTIFIED BY '비밀번호' WITH GRANT OPTION; (root 로 로그인)
* 데이터베이스와 테이블도 생성
  + 각 DB별, 사용자별, GRANT 쿼리를 이용하여 권한 부여해야 DB 접속가능.

MariaDB 포트 오픈:

* MariaDB 용 포트 열어주기iptables -I INPUT 1 -p tcp --dport 3306 -j ACCEPT로 3306포트도 열어준다 Maria DB는 3306포트를 사용한다.

<Tomcat - MariaDB / Java - MariaDB 연동>

연동 프로그램 설치 및 설정:

* tomcat-mariadb,java-mariadb를 연동 시켜 주기 위해서 오라클에서 리눅스용 sql connector.jar을 다운 받아 자바폴더와 MariaDB폴더에 넣어 연결.

연결된 DBMS의 로그 파일을 모니터링 및 관리:

* 로그 기록 활성화 설정:
  + 데이터베이스에 들어오는 모든 쿼리요청은 이곳에 기록되고 조회 가능
  + 일반로그는 TABLE과 FILE 방식으로 조회 할 수 있음.
  + 로그기록 활성화 및 저장방식 설정
    - show variables where Variable\_name in (‘version’, ‘log’, ‘general\_log’);

(값이 0 “로그 기록 안함” → set global general\_log = 1; “로그 기록 함”)

* + - 테이블 형식 저장

set global log\_output=’TABLE’;

* 에러 로그 저장경로 재설정:
  + 에러로그는 데이터베이스 구동, 연결 등 시스템적인 오류를 표시해줌.
  + 에러로그.는 초기설정 경로
    - var/lib/mysql/localhost.localdoain.err로 연결 되어 있음
  + 경로 재설정
    - 파일 생성: var/log/mariadb/error.log
    - 경로 재설정: etc/my.conf를 편집기 이용

<HeidiSQL:>

HeidiSQL 사용 배경:

* 데이터 베이스에 접속하여 명령하는 프로그램.

HeidiSQL 이용 MariaDB 접속:

* 마리아DB는 IP adress, MariaDB 사용자의 비밀번호를 입력하여 접속 및 연결한다.
* 원격이용자 SQL 이용 데이터 입력 및 조작 가능.

<WINscp>

WINscp 사용 배경:

* 자신의 컴퓨터에서 상대방 혹은 가상머신으로 파일을 옮겨주는 프로그램
* 미니프로젝트에서는 tomcat에 보낼 war파일을 보낼 용도로 사용할 예정이다

WINscp 사용 방법:

* WINscp공홈에서 다운
* putty와 같이 ip를 입력하고 포트는 22포트를 사용한다

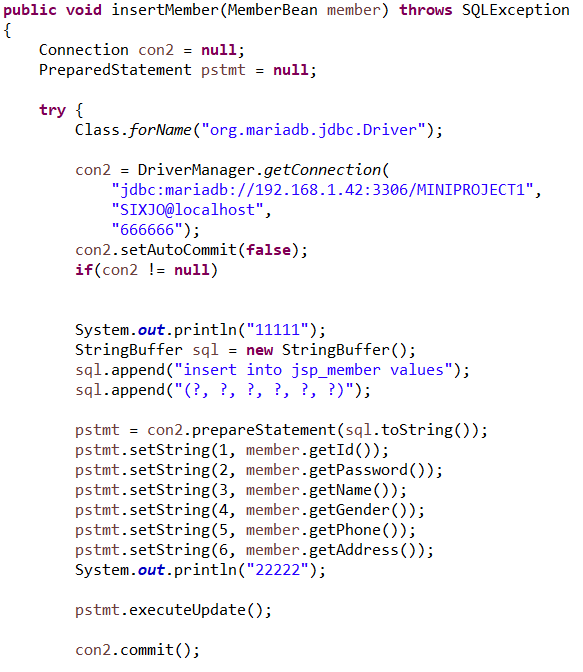
**JAVA 개발 Part 1:**

<회원가입 기능 구현>

JDBC 사용 배경:

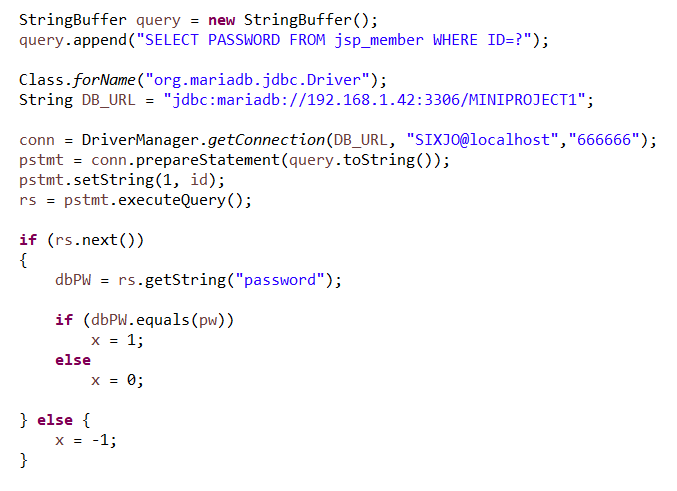
* JAVA에서 DB에 접속 할 수 있게 하는 자바 API임
  + DB에서 자료를 Query 하거나 업데이트 하는 방법을 제공

Maria DB 회원 데이터 테이블 생성 및 저장:



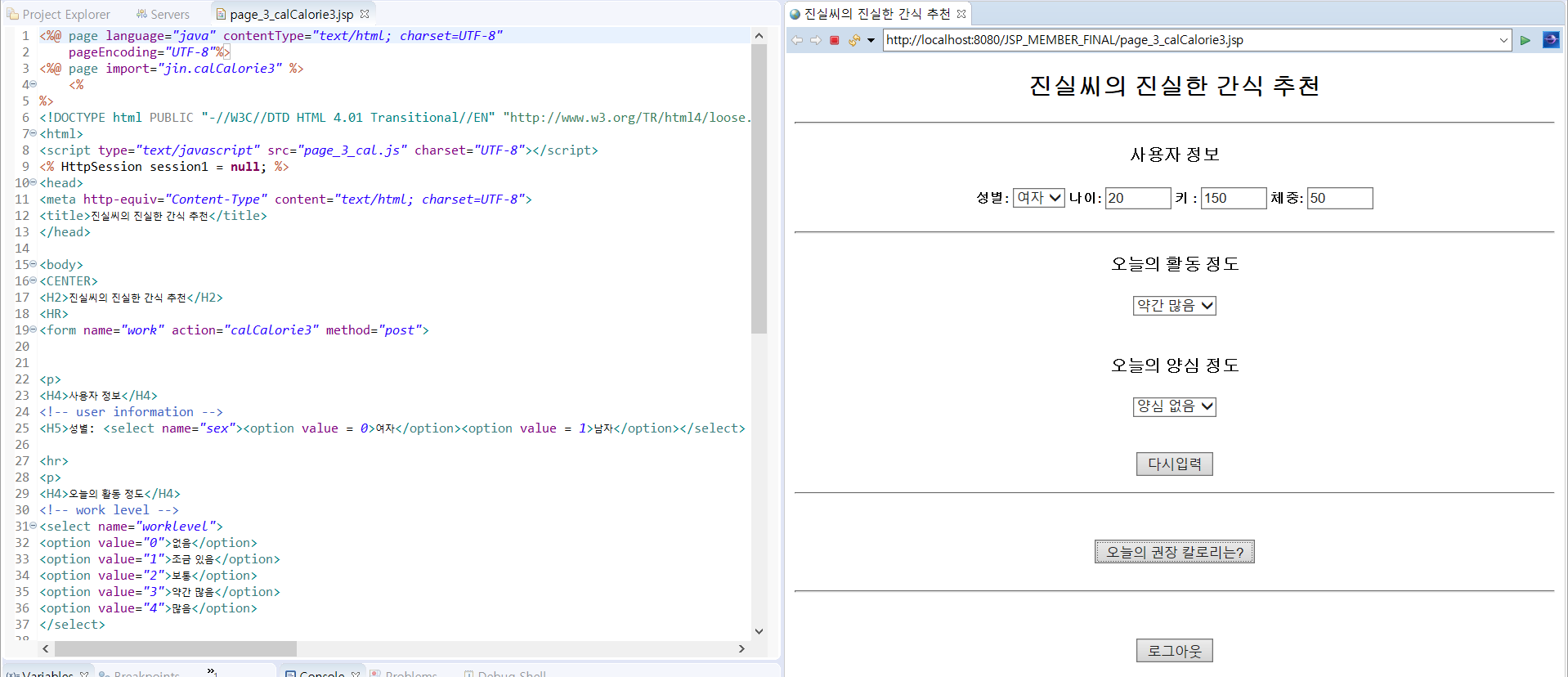
<로그인 기능 구현>

Maria DB 에 저장되어 있는 데이터, 비교 및 로그인 승인:



**JAVA 개발 Part 2:**

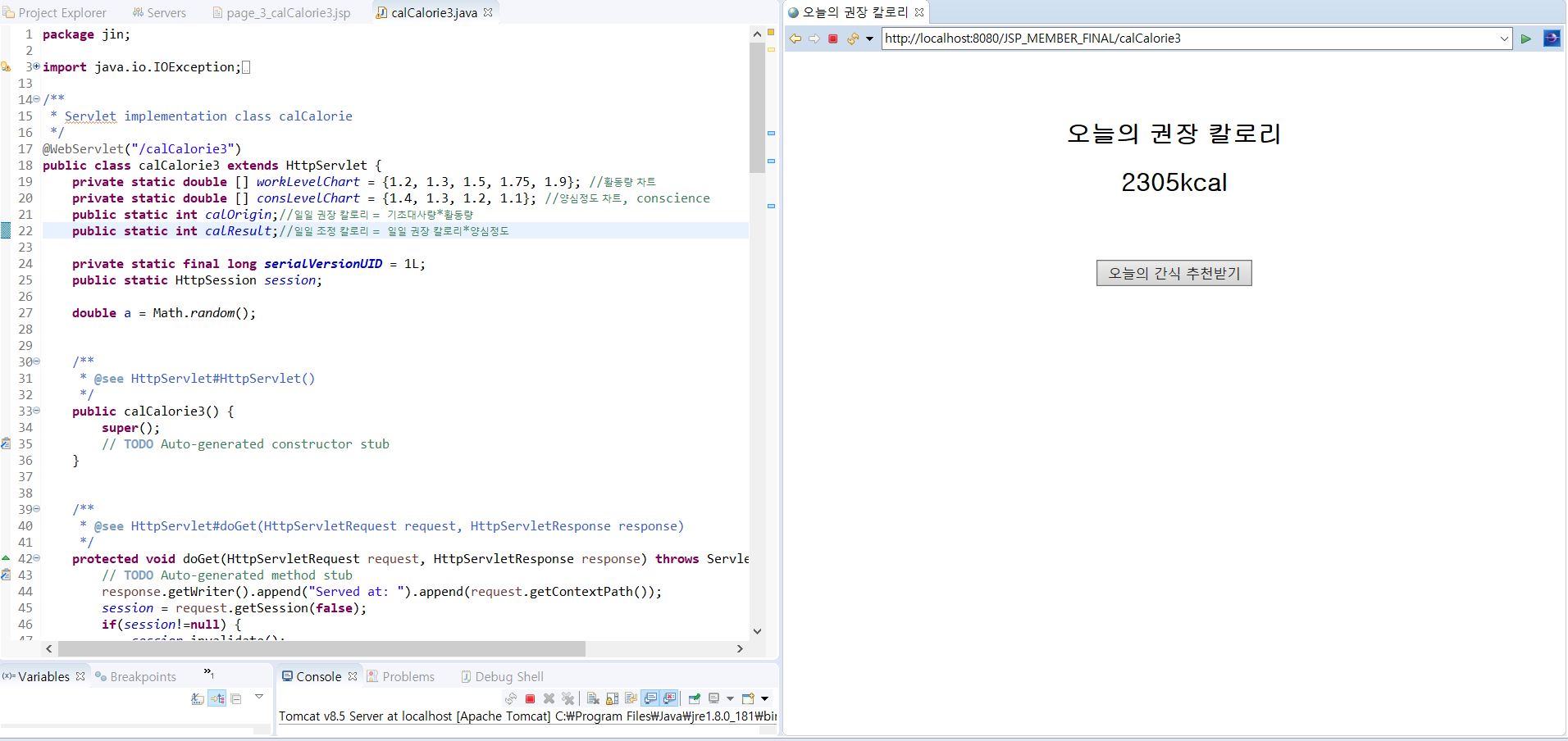
<사용자 신체정보 및 활동,양심 input 값 수집>



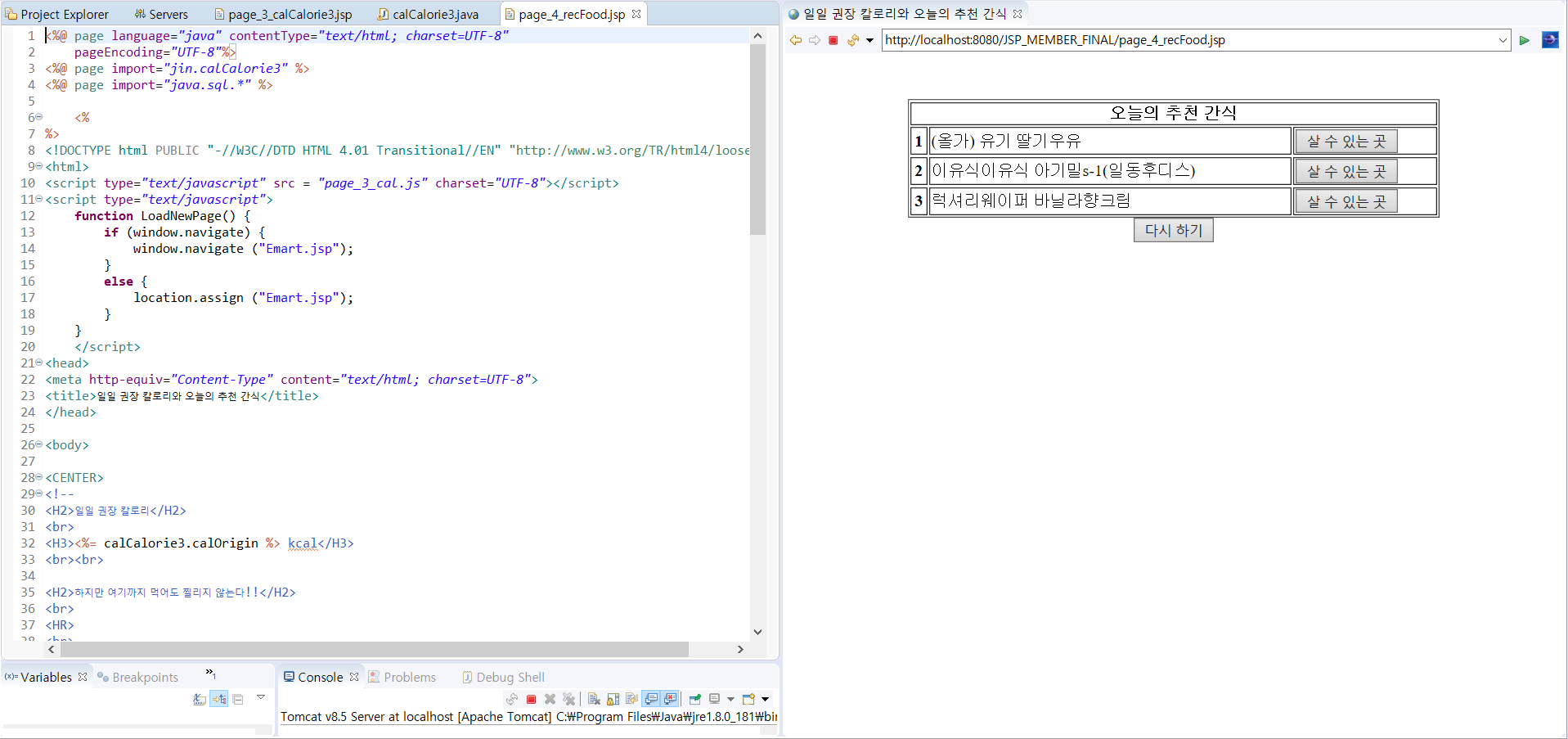
<JAVA구현 수식으로 권장 칼로리 및 가용 칼로리 계산>

권장 칼로리는 출력 후 저장되어 다음 페이지에서 사용되며

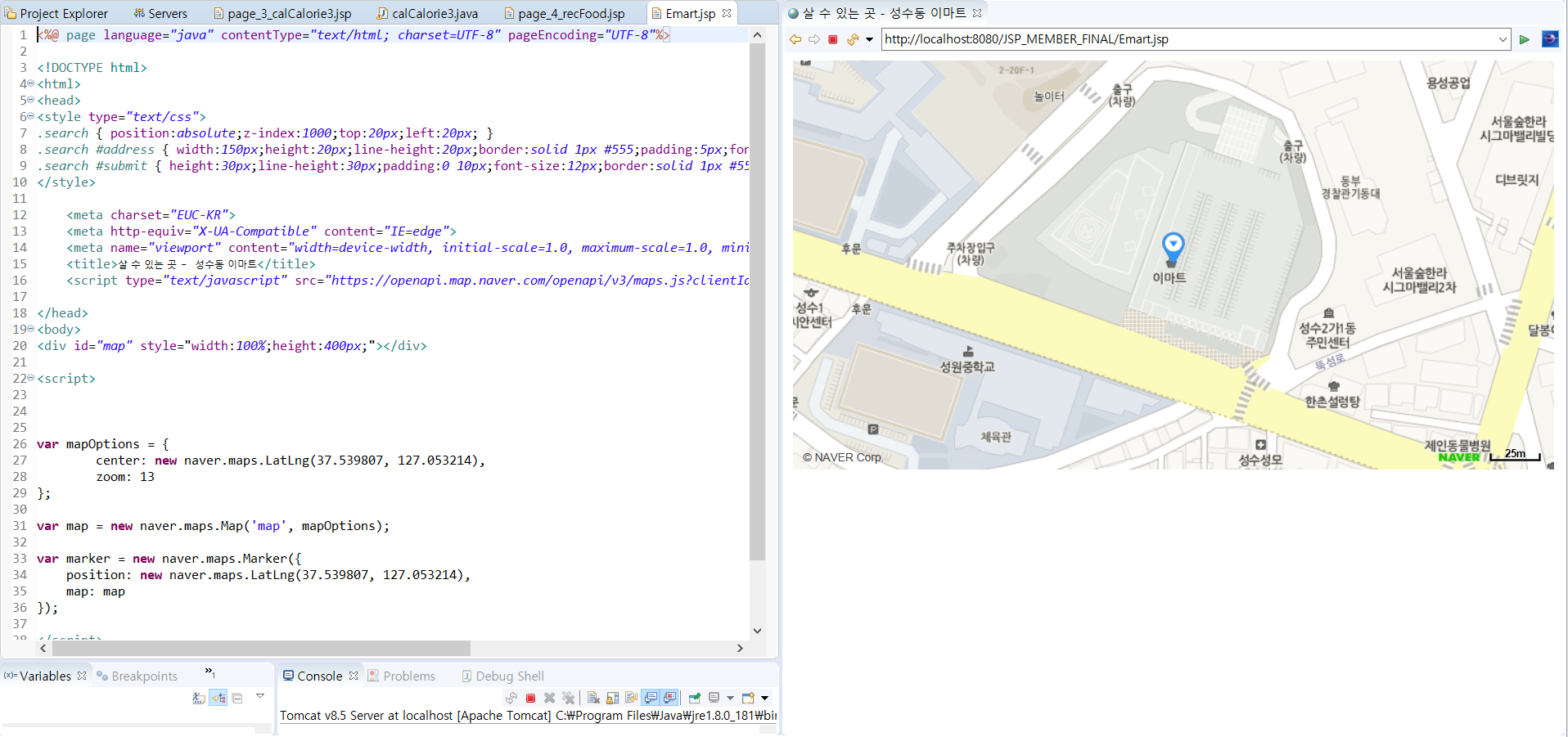
가용 칼로리는 출력 되지 않고 저장되어 다음 페이지에서 사용된다



<계산된 칼로리 이용, DB 연결하여 가용 칼로리 범위 내 간식 Random 3 추천>



**API 개발 Part:**

****

네이버 맵 API를 이용하여 연동

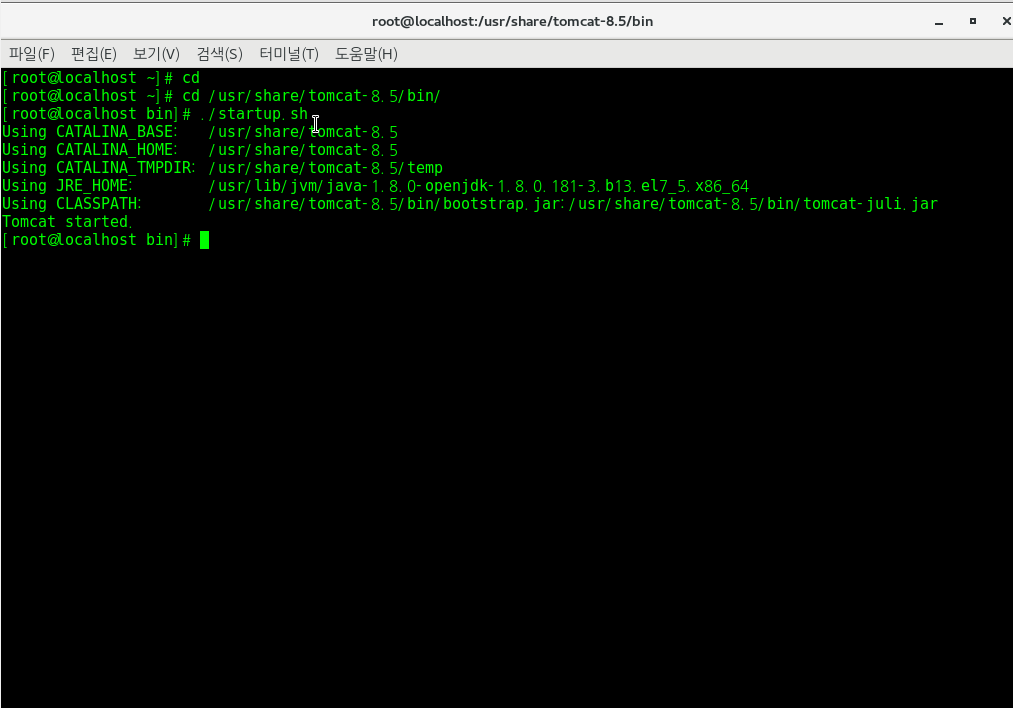
**DB 모델링 Part**

데이터 테이블 모델링:

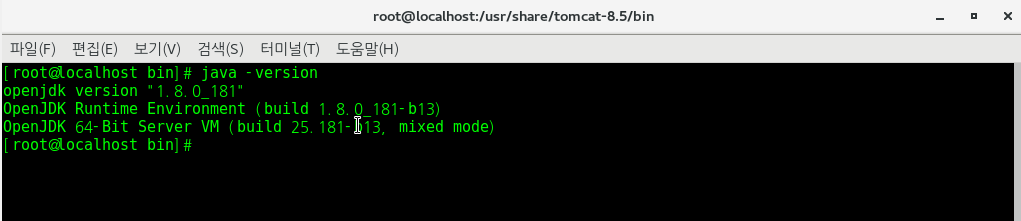
* HeidiSQL 이용, Server Mariad DB에 Data 테이블 생성
  + Member Table
  + Food\_info Table
* 각 테이블 당 Data Type 설정
  + Data Physical & Logical Architecture

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 회원   |  | | --- | | 회원번호 | | 로그인ID  패스워드  성별  생년월일  신장  체중  현재지역 | | MEMBER   |  | | --- | | MEMBER\_NO: VARCHAR2(6) | | id : VARCHAR(50)  password : VARCHAR(30)  gender : VARCHAR(10)  phone : VARCHAR(10)  address : VARCHAR(255) | |
| 음식   |  | | --- | | 음식 이름 | | 칼로리 | | FOOD   |  | | --- | | FOOD\_NAME : VARCHAR(30) | | FOOD\_CAL : INT(5) | |
| 운동   |  | | --- | | 운동 이름 | | 시간당 칼로리 | | WORKOUT   |  | | --- | | WORK\_NAME : VARCHAR(30) | | WORK\_CAL\_HR : INT(5) | |

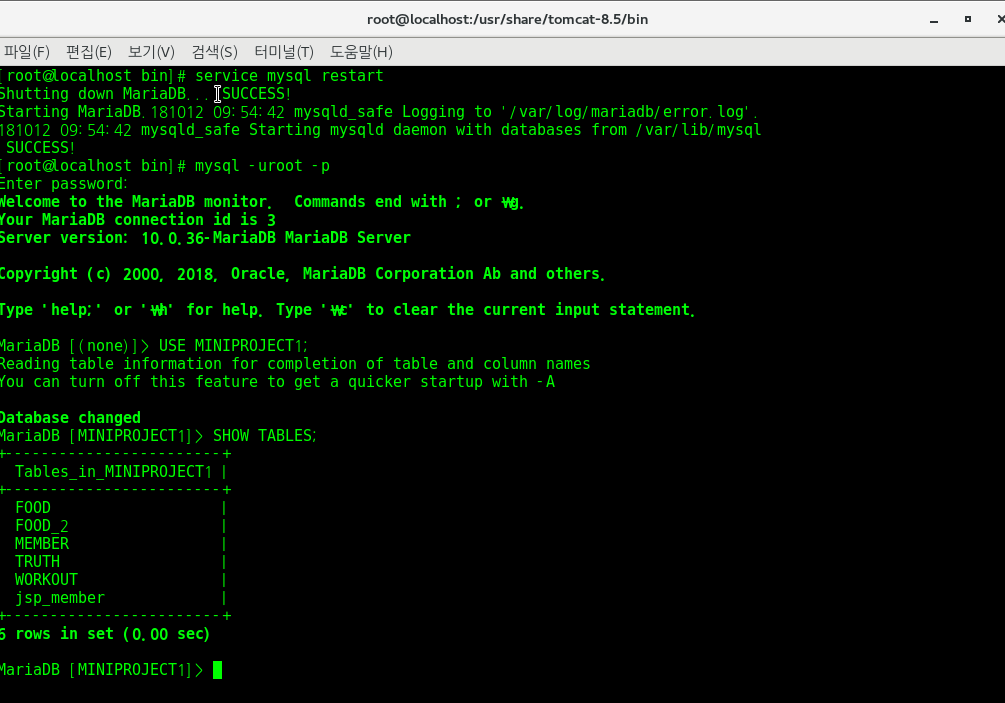
**6. 결과물**



톰캣 설치/실행 및 JRE연결 완료

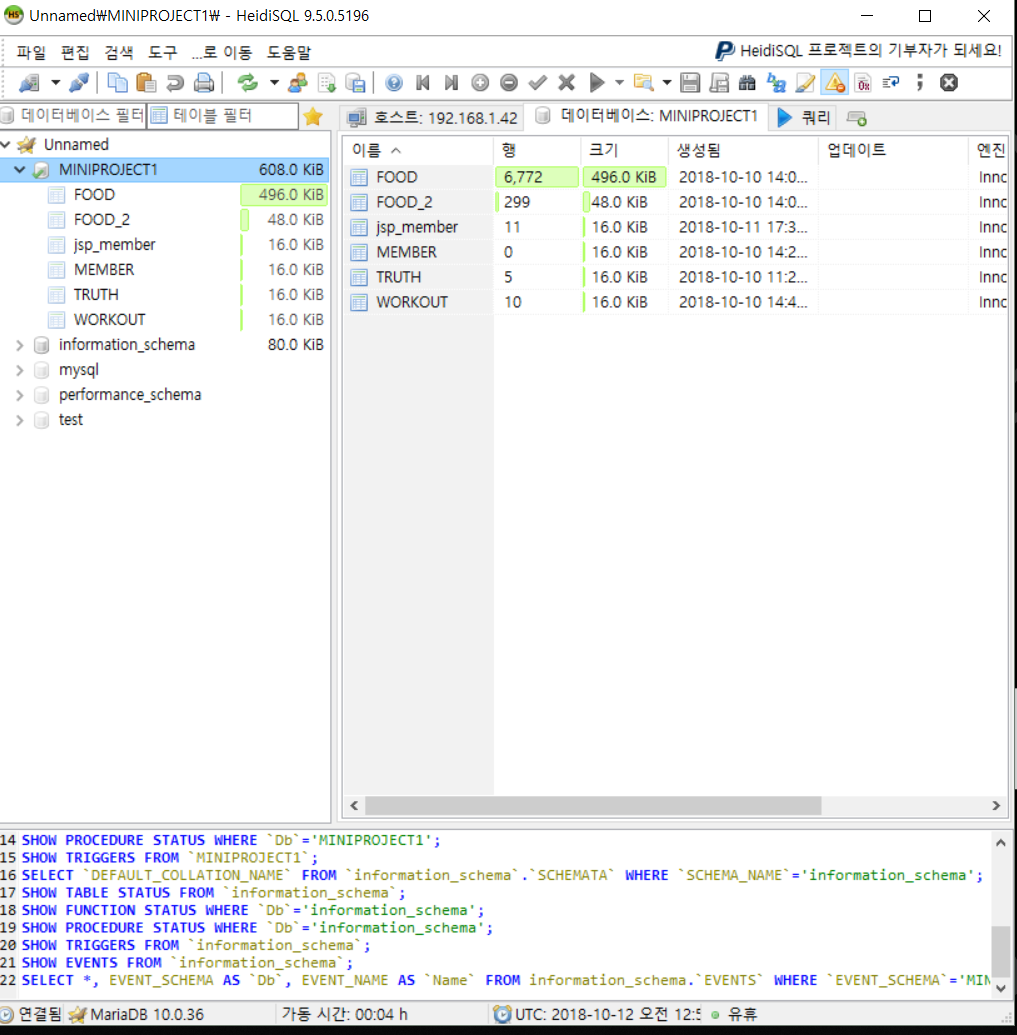


자바 설치 완료 및 버전확인

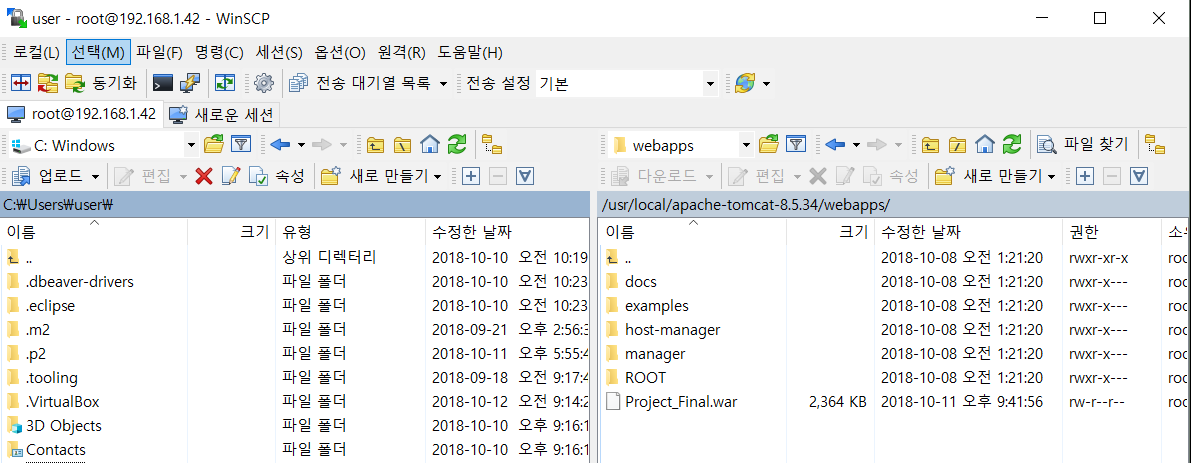


DBMS구동 완료

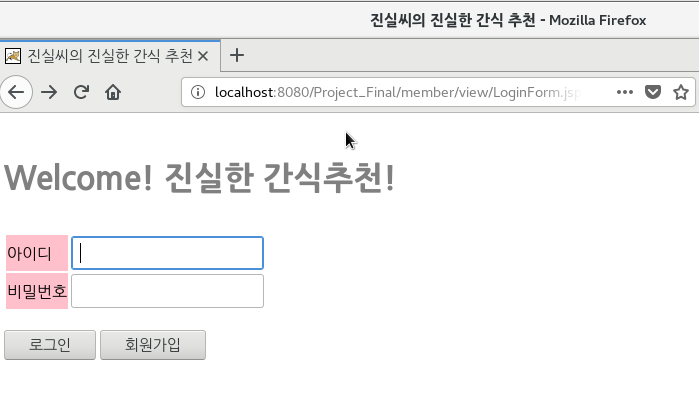
MINIPROJECT1 데이터베이스 및 하위 TABLE 설정 완료



서버컴퓨터 이외 컴퓨터에서 HEIDI SQL를 통하여 서버컴퓨터에 연결완료

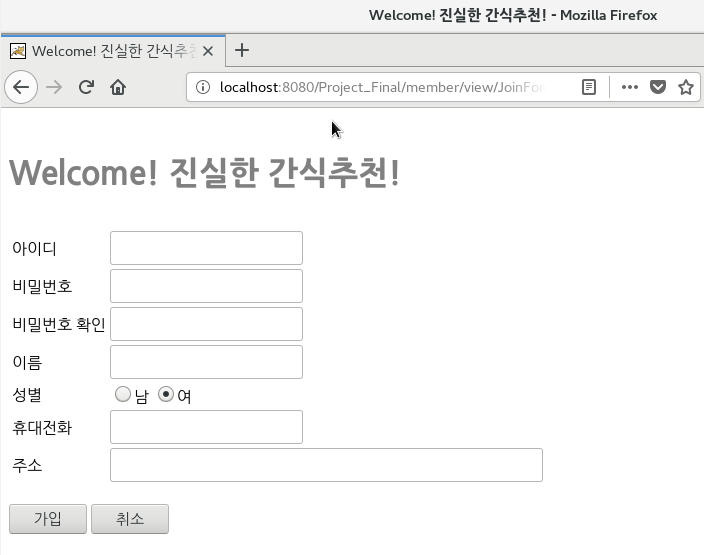


WINscp를 이용하여 webapps폴더에 파일 전송



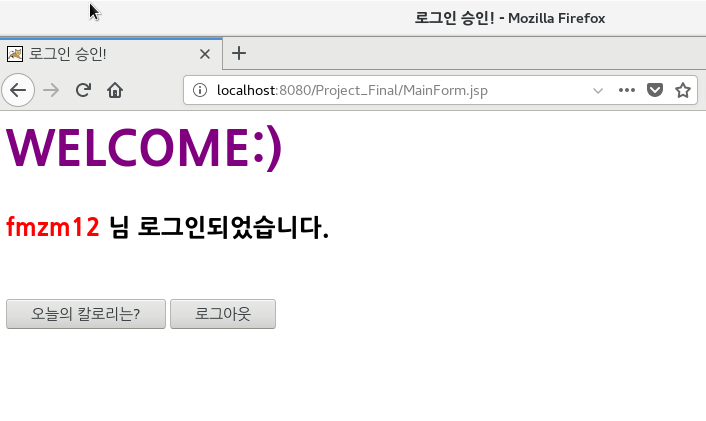
로그인 페이지

(DBMS에 저장된 회원 정보를 불러옵니다)



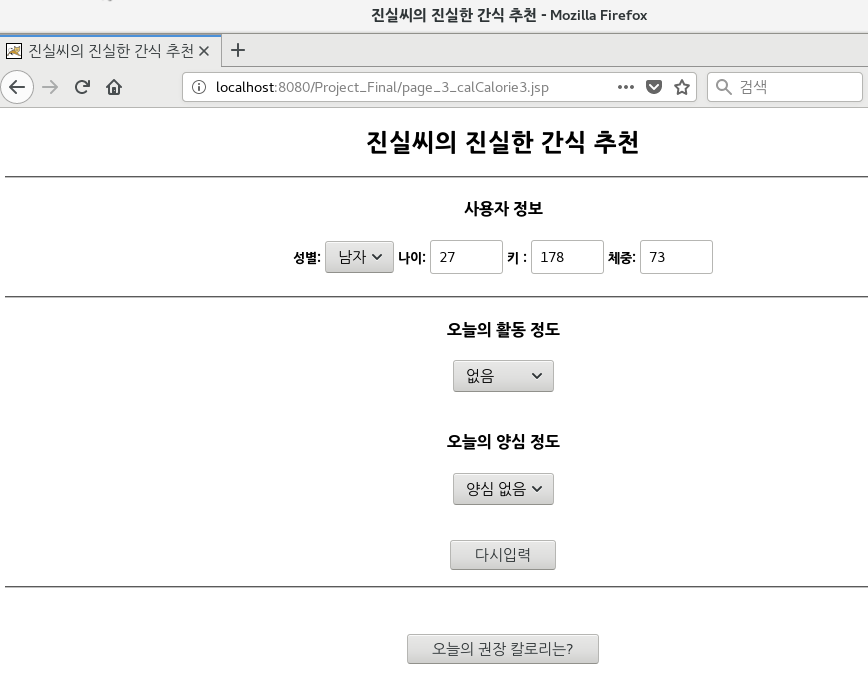
회원가입 페이지

(정보를 입력하고 가입을 누르면 DBMS에 저장됩니다)



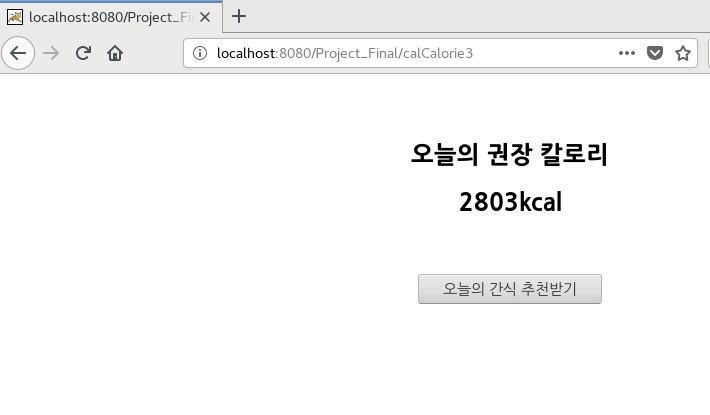
로그인 완료

(DBMS에서 저장된 아이디 값을 불러옵니다)



사용자 정보입력

(입력시 각 값들은 계산식의 변수로 사용됩니다)



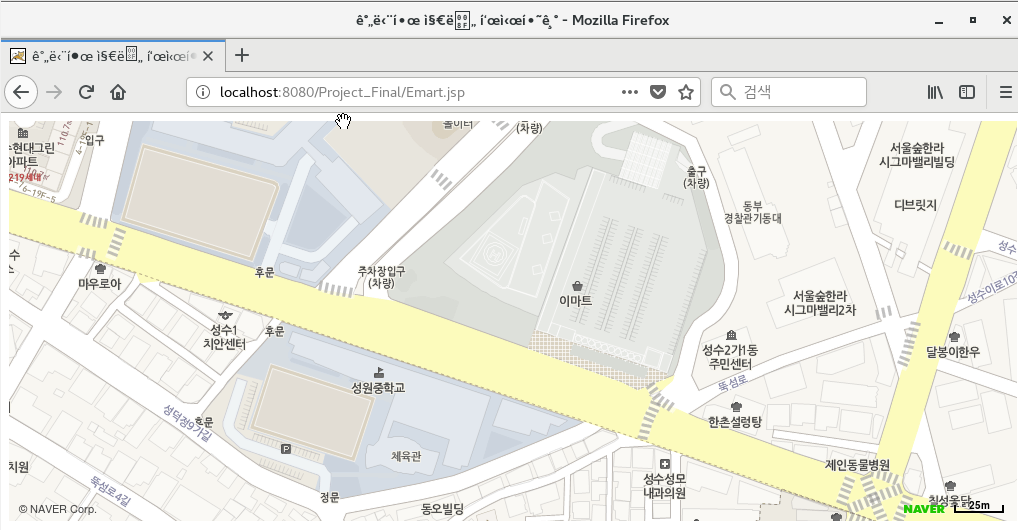
칼로리 계산

(계산된 값을 출력합니다)



추천간식

(칼로리 범위에 맞는 음식을 DBMS에서 꺼내어 출력합니다)



살 수 있는곳

(위의 물건들을 살 수 있는 곳을 정확히 추천해 줍니다)

**7. 향후 개발계획**

위치기반 서비스 추가

* 사용자가 회원가입 시 입력한 주소에 기반하여 추천된 간식을 구매할 수 있는 장소를 추가 할 계획
* 현재는 기간 관계상 구현해내지 못함

운동추천 및 운동 동영상 제공 시버스 추가

* 사용자가 추천된 간식에 불복하고 더 많은 칼로리 섭취를 원할 경우
* 더 많은 칼로리를 가진 음식을 추천하고 칼로리를 상쇄 할 수 있는 운동을 추천
* 운동은 DBMS에 있는 운동 데이터에 기반하여 맞춤 운동의 이름을 보여주고 YouTube API를 이용하여 바로 따라서 할 수 있도록 구현할 예정

아이디 중복확인 기능 및 회원탈퇴 기능 추가

* 회원가입시 아이디 중복확인 기능을 추가하여 같은 아이디가 두 번 이상 DBMS에 들어가지 않도록 확인
* DBMS에서 회원 정보를 제거 할 수 있도록 회원탈퇴 기능 추가