[Database system term_project]

Dr. HYU 앱 개발

컴퓨터소프트웨어

2015005141 최연수

1. 개발환경

Visual studio code

Docker version 19.03.2

Flask 1.1.1,

2. 사용언어

Python 3.7.0, HTML, Java-script

3. 실행방법

3.1 db_term_p 폴더에 들어간다

3.2 "docker-compose up -d" 명령어 입력

```
yeonsooui-MacBookPro:test yeonsoochoi$ ls
db_term_p
yeonsooui-MacBookPro:test yeonsoochoi$ cd db_term_p/
yeonsooui-MacBookPro:db_term_p yeonsoochoi$ ls
LICENSE                conf.d                 database.sql           docker-compose.yml      src
yeonsooui-MacBookPro:db_term_p yeonsoochoi$ docker-compose up -d
Starting db_term_p_postgres_1 ... done
Creating db_term_p_flaskapp_1 ... done
Creating db_term_p_nginx_1    ... done
```

3.3 database.sql 을 docker cp 명령어를 이용하여 postgres 컨테이너로 옮겨준다.

```
yeonsooui-MacBookPro:db_term_p yeonsoochoi$ docker ps
CONTAINER ID        IMAGE                  COMMAND                  CREATED              STATUS              PORTS                     NAMES
49610075d00e        nginx                  "nginx -g 'daemon of…"   47 seconds ago       Up 45 seconds       0.0.0.0:8080->80/tcp      db_term_p_nginx_1
3587d9a67c44        dreamwayjgs/dbcourse   "/bin/bash"              48 seconds ago       Up 46 seconds       80/tcp                    db_term_p_flaskapp_1
220bf8c941f7        postgres               "docker-entrypoint.s…"   About a minute ago   Up 47 seconds       0.0.0.0:54321->5432/tcp   db_term_p_postgres_1
yeonsooui-MacBookPro:db_term_p yeonsoochoi$ docker cp /Users/yeonsoochoi/Desktop/test/db_term_p/database.sql db_term_p_postgres_1:/
yeonsooui-MacBookPro:db_term_p yeonsoochoi$
```

3.4 Postgres 컨테이너 실행 후 "psql -f database.sql -U dbuser -d dbapp"을 입력하여 database.sql에 있는 내용을 해당 컨테이너의 db로 옮겨준다

```
yeonsooui-MacBookPro:db_term_p yeonsoochoi$ docker exec -it 220 /bin/bash
root@220bf8c941f7:/# ls
bin   database.sql  docker-entrypoint-initdb.d  etc   lib    media  opt   root  sbin  sys  usr
boot  dev           docker-entrypoint.sh        home  lib64  mnt    proc  run   srv   tmp  var
root@220bf8c941f7:/# psql -f database.sql -U dbuser -d dbapp
```

3.5 dreamwayjgs/dbcourse 컨테이너를 실행시킨다.

3.6 home/dbcourse_projects/src/flask 로 이동한 후 app.py를 확인한다.

3.7 라이브러리를 설치해준다.

pip install flask

pip install requests

3.8 Python app.py로 파일을 실행시켜준다.

3.9 http://127.0.0.1:8080 로 접근하여 처음 화면을 확인한다.

4. Table 설명

4.1 hosp_api

'공공 데이터 포털'에서 불러온 병원 api를 저장하는 table.

병원 명, 전문의 수, 주소, 위도, 경도 정보를 받아온다.

Primary key로 hid를 가지고 있다.

운영 시간을 저장하기위해, 월요일 시작 시간부터 일요일 끝나는 시간까지 정보를 저장한다.

초기 생성 값은, 주중 주말 관계없이 08시 개점, 17시마감으로 한다.

4.2 phar_api

'공공 데이터 포털'에서 불러온 약국 api를 저장하는 table.

약국 명, 위도, 경도 정보를 받아온다.

Primary key로 pid를 가지고 있다.

처방 가능 여부를 저장하기 위한 column을 가지고있다. 초기 값은 't'로 처방 가능을 나타낸다.

4.3 hospital

병원으로 등록되어있는 사람들을 저장하는 table.

사용자의 아이디, 비밀번호, 전화번호, 위치 정보를 가지고있다.

아이디를 primary key로 지정하여, 동일 아이디로 회원가입을 불가능하게 한다.

시리얼 번호로 personid를 가지고있다.

4.4 patients

환자로 등록되어있는 사람들을 저장하는 table.

사용자의 아이디, 비밀번호, 전화번호, 위치 정보를 가지고있다.

아이디를 primary key로 지정하여, 동일 아이디로 회원가입을 불가능하게 한다.

시리얼 번호로 personid를 가지고있다.

4.5 pharmacy

약국으로 등록되어있는 사람들을 저장하는 table.

사용자의 아이디, 비밀번호, 전화번호, 위치 정보를 가지고있다.

아이디를 primary key로 지정하여, 동일 아이디로 회원가입을 불가능하게 한다.

시리얼 번호로 personid를 가지고있다.

4.6 hosp_reser

병원 예약의 정보를 저장하는 table.

병원의 이름, 예약자의 이름,전화번호,예약한 날짜와 시간,예약한 사람의 personid 그리고 처방 여부를 저장한다.

처방 여부 저장을 위한 prescription column은 초기 'f'로 초기화 되어있고 이는 아직 처방하지 않았음을 뜻한다.

Primary key로 rid를 가지고 있다.

4.7 store_reser

약국 예약의 정보를 저장하는 table.

약국의 이름, 예약자의 이름,전화번호,예약한 날짜와 시간,예약한 사람의 personid 그리고 처방 여부를 저장한다.

처방 여부 저장을 위한 prescription column은 초기 'f'로 초기화 되어있고 이는 아직 처방하지 않았음을 뜻한다.

Primary key로 rid를 가지고 있다.

4.8 prescription

처방전의 내용을 저장하는 table.

(date, name , hosp_name, phar_name, medi_date) column으로 처방전의 기본 정보를 저장한다.

이때 date는 병원에서 처방 한 날짜. medi_date는 약국에서 처방한 날짜를 의미한다.

처방약은 총 세 개까지 저장이 가능하고, 각각 1회 투약 량, 1일 투여 횟수, 총 투약일 수를 저장한다.

etc column을 이용하여 처방의 변경, 수정, 확인, 대체 시 작성된 내용을 저장한다.

4.9 star

즐겨 찾는 병원 목록을 저장하는 table.

병원 이름과 그 병원을 즐겨 찾는 병원으로 등록한 환자의 personid를 저장한다.

4.10 subjects

병원의 진료과목을 저장하는 table.

병원의 이름과 그 병원이 진료하는 진료 과목을 저장한다.

5. 시나리오

5.1 회원가입 [동영상]

로그인 정보를 미리 등록 해놓지 않고, 직접 회원가입을 하도록 구현하였다.

로그인 정보가 없을 시, 회원가입 페이지로 이동하도록 하였고 회원가입 페이지 이동 버튼 또한 구현하였다.

회원가입 시 병원, 환자, 상점 은 중복 선택이 가능하도록 하였다.

회원가입 시 전화번호가 정규표현식이 아니라면 등록을 할 수 없도록 하였다.

5.2 로그인 [동영상]

병원, 환자, 상점 중 하나의 역할로만 로그인할 수 있다.

각 역할로 로그인 할 시, 해당 화면만을 볼 수 있다.

5.3 DB 구축

한양대를 기준점으로 하여 주변 5km의 병원과 약국을 기초 DB로 구현하였다.

5.4 환자 [동영상]

개인 정보 : 자주 가는 병원은 환자가 직접 즐겨 찾는 병원으로 등록할 수 있다.

최근 간 병원은 하나의 목록을 보여주며, 최근 이라는 기준은 병원 예약일 기준으로 한다. 예를 들어, 12월13일에 A병원을 예약하고 그 다음 12월14일에 B병원을 예약했다면 최근 간 병원은 B병원이 된다.

예약은 환자가 직접 시간과 날짜를 정할 수 있도록 하였다. 선택한 시간이 병원 운영시간이 아닐 경우, 다시 선택하도록 하였다. 시간 선택 시 문자가 입력되거나 24시간 60분 형식을 벗어나는 시간이 입력이 되어도 다시 선택하도록 하였다.

병원 검색 : 이름과 진료 과목으로 검색이 가능하도록 구현하였다. 이름 검색은 일부만 검색해도 검색이 가능하도록 하였고, 진료과목은 병원에서 설정 가능하다.

지도 위에 병원 마크를 표시하였으며, 현재가 진료시간이 아니면 검정 마크, 현재가 진료시간이면 빨간 색 마크가 뜨도록 구현하였다.

상점 검색 : radio-box에 약국을 선택하고 검색하면 약국만 검색되도록 하였다. 또한 약국 선택 시 지도에선 약국 마크만 생성되도록 하였다. 약국 처방 가능 여부는 가능과 불가능으로 구분하였으며 가능 여부는 약국에서 설정할 수 있다.

5.5 병원 [동영상]

병원 정보 : 병원 정보는 환자 로그인 화면에서 볼 수 있으며, 병원 이름, 주소, 전문의 의원 수, 진료 과목을 알 수 있다. 처음 병원으로 로그인 시 어느 병원으로 로그인할지 선택할 수 있고, 해당 병원의 진료과목과 진료 시간을 수정할 수 있다.

환자 정보 : 이 병원을 예약한 사람들의 예약 정보를 볼 수 있고 이를 처방하거나 삭제할 수 있다. 처방하기를 click시 처방전을 작성할 수 있게 된다. 병원 이름, 환자 이름, 처방 날짜는 자동으로 기입되어 있다. 이는 prescription table에 저장된다.

처방을 완료하면, 처방이 완료되었음을 알 수 있다.

5.6 상점 [동영상]

처음 로그인 시 어느 약국인지 선택할 수 있다.

환자가 예약하면 그 예약 정보를 볼 수 있다.

약국에선 처방이 가능한지 불가능한지 선택할 수 있다.

상점에선 예약 내역을 삭제할 수 있다.