

모두의 카페 탐방기,

Cafestagram

- 서울시 구별 베스트카페 소개

- 홈페이지 주제
- 구성 요소
- 구현 내용 및 핵심 기술
- 개선점

1710222 임수영

홈페이지 주제 >

회원가입 후 본인 계정으로 로그인하여 자신이 방문했던 서울카페의 후기를 포스팅하는 웹페이지이다. 포스팅 시 페이지 하단부의 버튼을 클릭하여 본인이 다녀온 카페를 구별로 분류하여 포스팅할 수 있는 기능을 갖추고 있다.

구성 요소 >

```
s1710222@whistle:~/public_html/project$ ls -l
total 52
-rwx---rwx 1 s1710222 students 582 Jun 21 14:11 account.txt
-rwx---rwx 1 s1710222 students 1065 Jun 21 12:21 check.php
-rwx---rwx 1 s1710222 students 303 Jun 21 14:10 check.txt
drwxr-xr-x 4 s1710222 students 4096 Jun 21 13:41 district
-rwx---rwx 1 s1710222 students 1077 Jun 21 14:23 home.html
-rwx---rwx 1 s1710222 students 816 Jun 21 12:18 home.php
-rw-r--r-- 1 s1710222 students 1486 Jun 21 06:33 main.html
drwxr-xr-x 2 s1710222 students 4096 Jun 20 02:53 pictures
-rw-r--r-- 1 s1710222 students 4680 Jun 21 12:59 posting.html
-rw-r--r-- 1 s1710222 students 2006 Jun 21 12:22 signup.html
-rwx---rwx 1 s1710222 students 855 Jun 21 12:20 signup.php
-rw-r--r-- 1 s1710222 students 1446 Jun 21 12:39 style.css
s1710222@whistle:~/public_html/project$
```

```
s1710222@whistle:~/public_html/project$ cd district
s1710222@whistle:~/public_html/project/district$ ls -l
total 108
-rw-r--r-- 1 s1710222 students 1047 Jun 21 13:41 1.php
-rw-r--r-- 1 s1710222 students 1048 Jun 21 13:38 10.php
-rw-r--r-- 1 s1710222 students 1049 Jun 21 13:38 11.php
-rw-r--r-- 1 s1710222 students 1046 Jun 21 13:38 12.php
-rw-r--r-- 1 s1710222 students 1051 Jun 21 13:37 14.php
-rw-r--r-- 1 s1710222 students 1047 Jun 21 13:36 15.php
-rw-r--r-- 1 s1710222 students 1050 Jun 21 13:36 16.php
-rw-r--r-- 1 s1710222 students 1050 Jun 21 13:35 17.php
-rw-r--r-- 1 s1710222 students 1048 Jun 21 13:35 18.php
-rw-r--r-- 1 s1710222 students 1051 Jun 21 13:35 19.php
-rw-r--r-- 1 s1710222 students 1050 Jun 21 13:41 2.php
-rw-r--r-- 1 s1710222 students 1054 Jun 21 13:34 20.php
-rw-r--r-- 1 s1710222 students 1048 Jun 21 13:33 21.php
-rw-r--r-- 1 s1710222 students 1051 Jun 21 13:33 22.php
-rw-r--r-- 1 s1710222 students 1048 Jun 21 13:33 23.php
-rw-r--r-- 1 s1710222 students 1046 Jun 21 13:32 24.php
-rw-r--r-- 1 s1710222 students 1049 Jun 21 13:32 25.php
-rw-r--r-- 1 s1710222 students 1048 Jun 21 13:40 3.php
-rw-r--r-- 1 s1710222 students 1048 Jun 21 13:40 4.php
-rw-r--r-- 1 s1710222 students 1047 Jun 21 13:40 5.php
-rw-r--r-- 1 s1710222 students 1049 Jun 21 13:40 6.php
-rw-r--r-- 1 s1710222 students 1045 Jun 21 13:39 7.php
-rw-r--r-- 1 s1710222 students 1050 Jun 21 13:39 8.php
-rw-r--r-- 1 s1710222 students 1046 Jun 21 13:39 9.php
drwxr-xr-x 2 s1710222 students 4096 Jun 21 02:52 map
drwx---wx 2 s1710222 students 4096 Jun 21 14:29 upload
s1710222@whistle:~/public_html/project/district$
```

구현 내용 및 핵심 기술 >

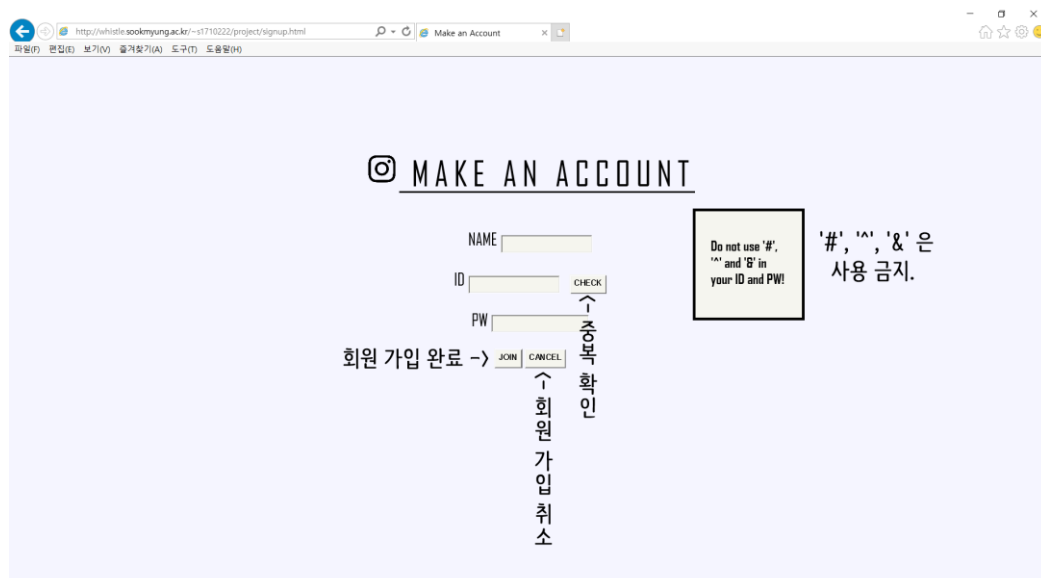
1. 회원가입 기능

main.html



위 화면은 홈페이지의 메인 페이지인 'main.html' 페이지이다. 본인 계정이 있는 경우 'SIGN IN' 버튼을 클릭하여 'home.html'로 이동해 포스팅을 진행할 수 있고, 계정이 없는 경우 'SIGN UP' 버튼을 클릭하여 'signup.html'로 이동해 회원가입을 진행하여 계정을 생성한다. 우선, 계정이 없다고 가정하고 'signup.html'로 이동하여 계정을 만들어 보자.

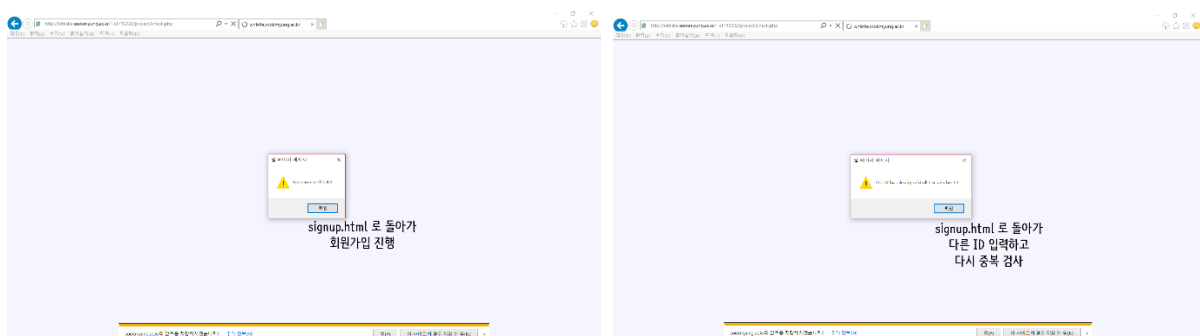
signup.html



‘main.html’에서 ‘SIGN UP’ 버튼을 클릭하게 되면 위와 같은 창으로 전환된다. ‘CHECK’ 버튼은 사용자가 ID 값을 입력했을 시 기존에 등록했던 ID와 같은 값이 있는지의 여부를 판단한다. 그 판단 여부 기능은 ‘JOIN’버튼의 기능과 비슷한데, ‘JOIN’ 버튼은 ‘CHECK’를 클릭하여 중복 검사를 한 후, 클릭해 ‘signup.php’ 로 이동하여 사용자가 입력했던 ID 값과 PW 해쉬값을 ‘account.txt’파일에 저장하여 ‘main.html’에서 로그인 시 계정의 존재 여부를 판별할 수 있게끔 한다. ‘CANCEL’ 버튼은 회원가입을 취소하고 다시 초기화면(‘main.html’)로 돌아가는 기능을 한다.

예시로 ‘123’이라는 ID 값을 등록하고자 한다고 가정하자. ‘123’을 입력하고 ‘CHECK’ 버튼을 누른다.

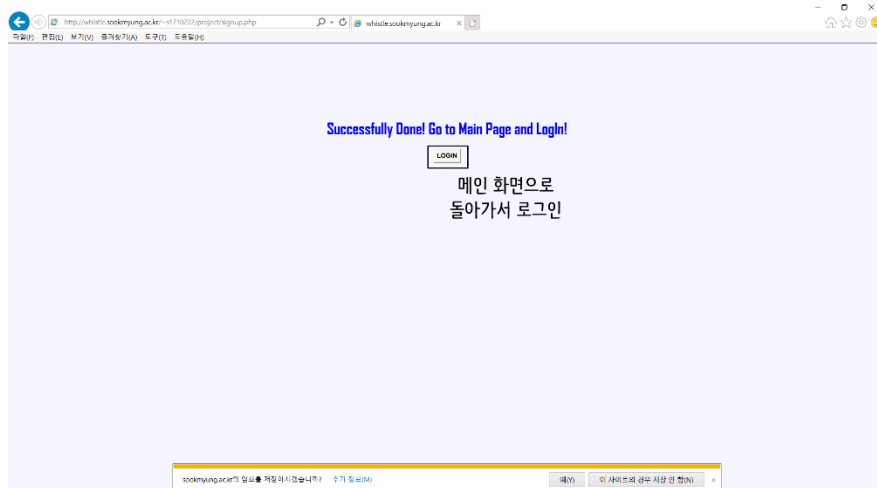
check.php



‘check.php’는 ‘check.txt’ 파일을 검사하여 기존에 등록되어 있는 ID 인지 아닌지를 판별한다. 만일 ‘123’이 기존에 존재하지 않는 ID 라면 (아무 ID 도 존재하지 않는 경우 포함) ‘check.txt’파일에 해당 ‘ID’를 fwrite 함수를 통해 저장한다. 이 때, 모든 계정이 텍스트 파일에 문자열 형식으로 입력되기 때문에 ‘#’ 문자로 분리하여 계정 존재 여부를 판별한다. 그렇기에 ‘signup.html’의 <div>태그 안 문장에서 해당 문자를 계정 생성시 사용하지 말라고 경고한 것이다. 그리고 ‘You can use this ID’라는 alert 메시지를 띄운 후 다시 ‘signup.html’ 파일로 돌아가 회원가입 절차를 마무리한다.

하지만 '123'이 기존에 존재하는 ID 라면 'This ID has already existed! Use another ID'라는 alert 메시지를 띄우게 된다. 이 경우, 해당 ID 는 'check.txt' 파일에 저장되지 않는다. 다시 'signup.html' 로 돌아가서 다른 ID 값을 입력하고 다시 중복 검사를 해야 되는 것이다.

signup.php



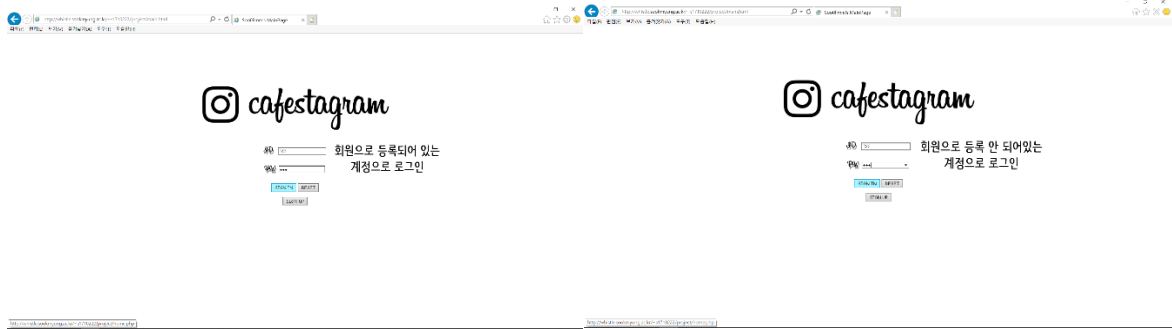
중복 검사를 완료한 계정은 'JOIN' 버튼을 클릭하여 'signup.php'로 이동해 사용자가 입력한 ID, 그리고 PW 의 해쉬값(ENC)을 'str'이라는 문자열에 "\$ID.#".\$ENC."&";" 형태로 저장하여 fwrite 함수를 통해 'account.txt' 파일에 문자열 형태로 저장(입력)한다. 그 후, 'LOGIN' 버튼을 클릭하여 메인 화면(main.html)으로 돌아가서 로그인을 실행한다.

2. 로그인 기능

'main.html'에서 'SIGN IN' 버튼을 클릭하면 'home.php' 페이지에서 기존에 등록했던 계정인지의 여부를 검사할 수 있다.

home.php 소스코드 전문

```
1 <html>
2 <head><link rel = stylesheet TYPE = "text/css" href = "style.css">
3 </head>
4 <?php
5     $ID = $_POST['ID'];
6     $ENC = $_POST['ENC'];
7     $str = $ID."#".$ENC;
8     $file = fopen('account.txt', 'r');
9     $filesize = filesize('account.txt');
10    $string = fread($file, $filesize);
11    $words = explode("&", $string);
12    foreach ($words as $word){
13        if ($word == $str){
14            fclose($file); ?>
15            <?php echo ("<script language = 'javascript'>document.location.href = 'home.html';</script>"); ?>
16        }
17    }
18    <?php fclose($file); ?>
19    <?php echo ("<script language = 'javascript'>alert('Check your EMAIL and PASSWORD!');</script>"); ?>
20    <?php echo ("<script language = 'javascript'>document.location.href = 'main.html';</script>"); ?>
21 </html>
```



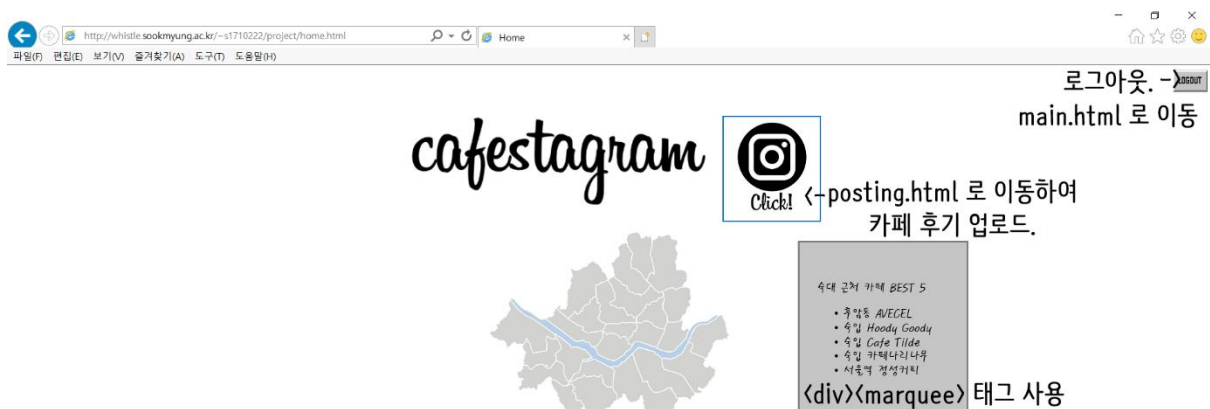
‘home.php’에는 ‘main.html’에서 사용자가 입력했던 ID와 PW의 해쉬값을 각각 ‘\$ID’와 ‘\$ENC’라는 변수에 저장한 후, 이를 ‘#’문자를 이용해 결합한 형태인 ‘\$str’의 값이 존재한다. 이 값과 ‘signup.php’에서 회원가입이 완료된 계정이 저장되어있는 ‘account.txt’파일을 읽어들이 ‘\$str’이 기존 계정에 존재하는지의 여부를 판단한다. 이 때, 텍스트파일에 저장되어있는 ID와 해쉬값은

ID#ENC&

형태로 저장되어 있으므로 explode 함수를 사용하여 ‘&’를 기준으로 ‘\$word’로 분리해내어 하나하나씩 ‘\$str’과 비교한다. 만일 ‘\$str’이 존재하는 경우, ‘account.txt’ 파일을 fclose 함수를 통해 닫고, ‘home.html’로 바로 이동한다. ‘\$str’이 존재하지 않는 경우 ‘Check your EMAIL and PASSWORD!’라는 alert 메시지를 띄운 후 ‘main.html’로 이동하여 위와 같은 작업을 반복한다.

3. 포스팅 기능

home.html

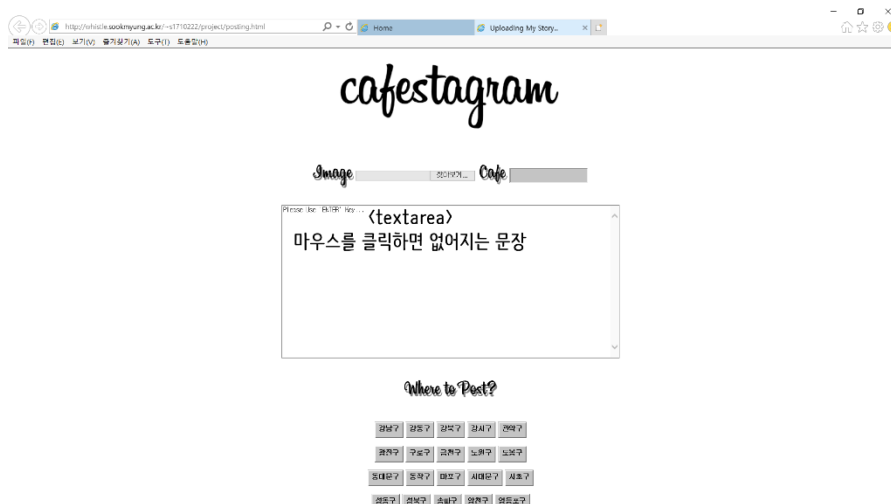


각각에 해당하는 php파일로 이동

25 Districts of Seoul				
강남구	강북구	강동구	강서구	관악구
광진구	구로구	금천구	노원구	도봉구
동대문구	동작구	마포구	서대문구	서초구
성북구	성북구	송파구	양천구	영등포구
용산구	윤문구	중랑구	종로구	홍원구

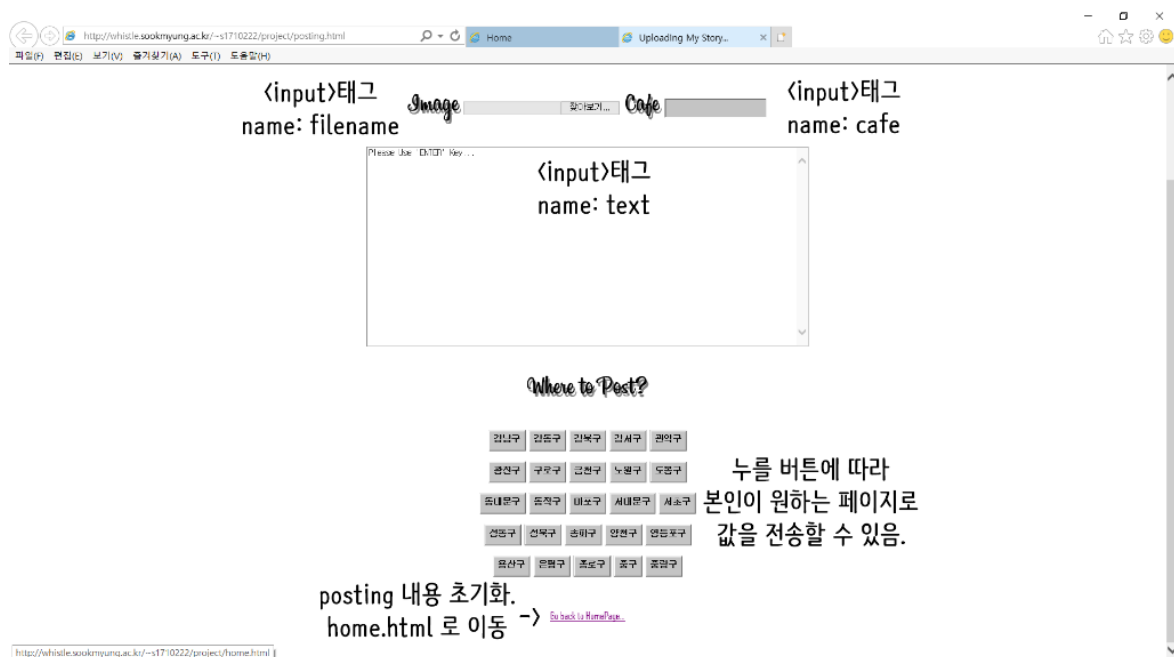
‘home.html’은 회원가입과 로그인에 성공하였을 경우 나타나는 페이지이다. 이 페이지는 메인 기능인 포스팅 기능 외에도 여러 기능을 함축하고 있는데, 우선 화면 우측 상단부의 ‘LOGOUT’ 버튼을 클릭하면 로그아웃되어 ‘main.html’으로 이동하고, <div>태그 안 내용은 <marquee>태그를 이용하여 ‘숙대 근처 카페 BEST 5’ 를 보여준다. 그리고 하단부 테이블에는 서울시 25 개 구 각각의 포스팅 php 파일로 이동할 수 있는 하이퍼링크를 걸어두었다. 페이지 로고 우측 ‘Click’ 버튼을 누르면 카페 후기를 포스팅할 수 있는 ‘posting.html’로 이동한다.

posting.html



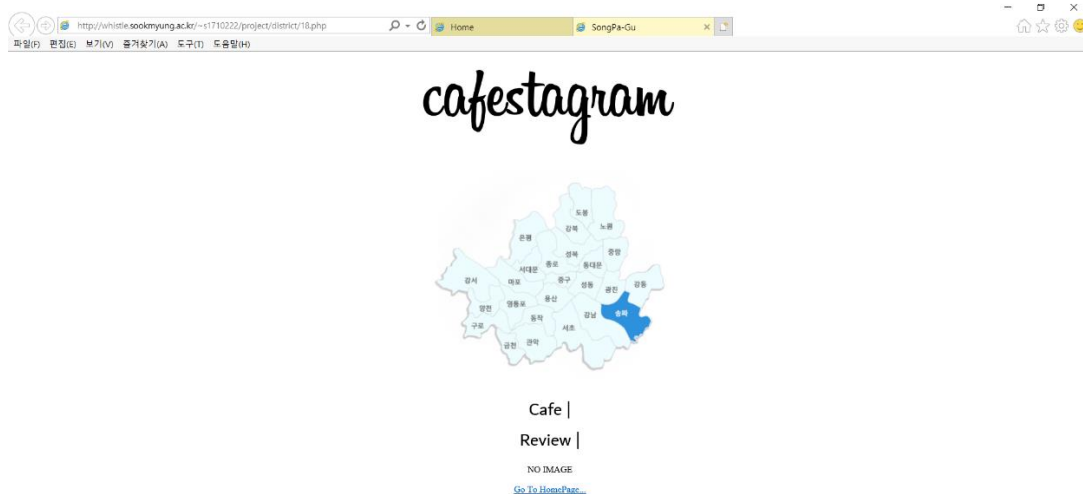
```
<textarea name = "text" rows = "20" cols = "100" onclick = "this.value=" "> Please Use 'ENTER' Key...</textarea>
```

<textarea>태그 속 내용을 마우스로 클릭하면 자동으로 내용이 전체삭제된다.



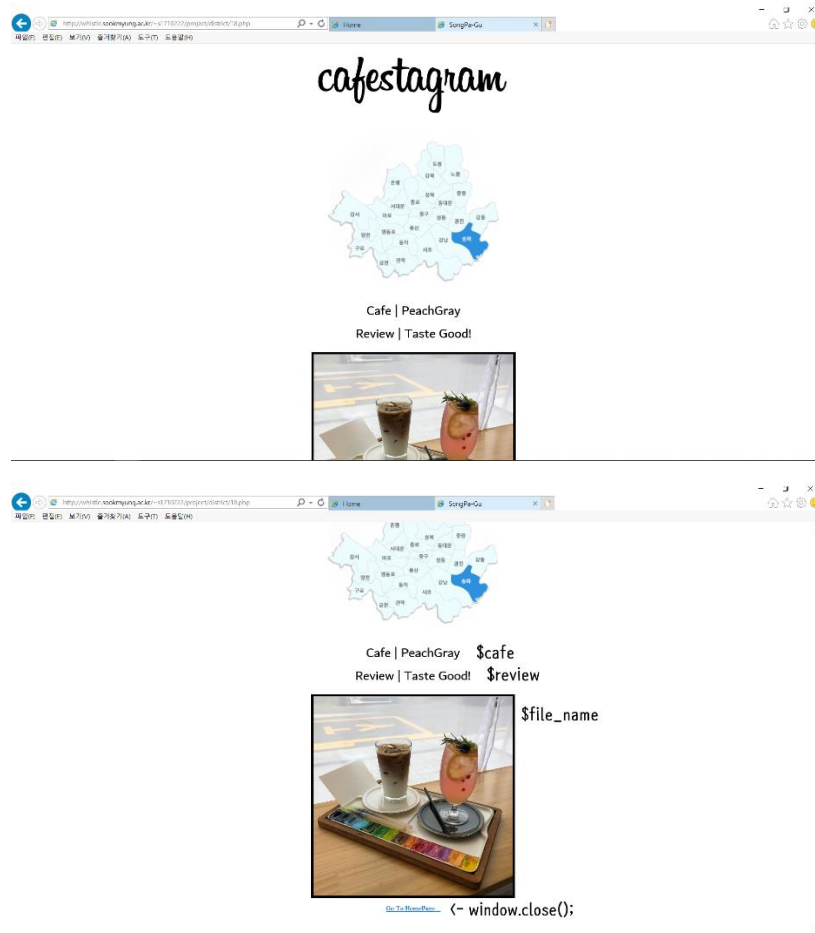
하단부 25 개의 버튼은 type 이 'submit'이 아닌 'button'으로 정의되어 있다. 이는 하나의 <form> 태그 안에서 해당 버튼에 onclick 이벤트를 걸어 자바스크립트 내에서 form 의 유효성 검사 후 마지막에 submit()을 수행하기 위함이다. 이렇게 하면 사용자가 원하는 php 페이지로 값을 전송할 수 있게 되는데, 포스팅 기능을 실행하기 전 각각의 페이지의 초기화면은 다음과 같다. 'home.html'의 테이블 속성 중 '송파구'를 눌렀을 시의 결과이다. 홈페이지 로고와 지도, 그리고 <input>태그의 'filename', 'café', 'text'의 NULL 값을 확인할 수 있다.

ex) district/18.php



posting.html 에서 위 페이지로 데이터를 전송해 보겠다.





```

1 <html>
2 <title>SongPa-Gu</title>
3 <body>
4 <p align = "center"><img src = "../pictures/cafestagram2.PNG"></p>
5 <p align = "center"><img src = "../map/18.png" width = "25%"></p>
6 <?php
7     $cafe = $_POST['cafe'];
8     $review = $_POST['text'];
9     echo "<p align = 'center'><font face = 'Dubai' size = '6'>Cafe | ".$cafe."<br>Review | ".$review."</font></p>";
10    $file_tmp_name = $_FILES['filename']['tmp_name'];
11    $file_name = $_FILES['filename']['name'];
12    $words = explode(".", $file_name);
13    $ext = $words[1];
14    if ($ext==NULL){
15        echo "<p align = 'center'>NO IMAGE</p>";
16    }
17    else if ($ext == "png" || $ext == "jpg" || $ext == "gif" || $ext == "PNG" || $ext == "JPG" || $ext == "GIF") {
18        $file_name = str_replace(".", "_", $file_name);
19        move_uploaded_file($file_tmp_name, './upload/'.$file_name);
20        echo "<p align = 'center'><img src = './upload/{\$file_name}' width = '25%' border = '5'></p>";
21    }
22    else{
23        echo "Upload Failed<br>";
24    }
25    echo "<p align = 'center'><a href = 'javascript: window.close();'>Go To HomePage...</a></p><br><br>";
26 >>
27 </body>
28 </html>

```

버튼을 클릭하여 원하는 페이지로의 이동이 성공적으로 이루어졌음을 알 수 있고, 사용자가 입력했던 입력값을 모두 php 파일에 반영하여 나타냄을 알 수 있다. 이 때, 업로드할 수 있는 파일은 이미지 파일로 한정했고, 이를 확장자를 통해 판별할 수 있게끔 코드를 작성하였다. 이미지 파일은

{원래파일명}_{사용자가 입력했던 카페명}.확장자

로 변경하여 district 디렉토리의 하위 폴더인 upload 폴더에 업로드하였다.

개선점 >

본디 프로젝트의 목적은 계정을 보유하고 있는 사용자에게 한해 접근이 가능한 페이지를 만들고 그 사용자들에 한해 위키피디아처럼 언제든지 내용을 수정할 수 있는, 오픈소스와 같은 커뮤니티를 만들고자 함이었다. 그리하여 웹페이지에 접속한 사용자들 모두에게 동일한 내용의 페이지를 만들고자 회원가입 및 로그인 시 각각 하나의 텍스트파일에 사용자 정보를 저장하였으며, 굳이 사용자 계정 하나하나에 고유한 텍스트 파일을 부여하지 않았다. 그런데 사용자가 입력한 데이터를 POST 방식으로 전송하고, 해당 페이지를 새로고침하거나 'home.html'에서 하이퍼링크를 통해 접속하면 초기화면으로 되돌아간다. 일회용 포스팅 프로그램이 돼 버린 셈이다. 사용자 정보를 담고 있는 각각의 텍스트 파일이 없어서 발생하는 문제인가 생각했지만, 서버 요청 시 웹 브라우저에 저장된 쿠키를 전달받아 그 값을 읽어내는 방식이 부재하기에 발생하였다. 따라서, 쿠키를 읽어서 가져오고, 그 값을 저장하는 로그인 기능과, 로그아웃 시 쿠키를 삭제하는 기능까지 구현한다면 조금 더 완성도 있는 프로그램이 될 것으로 예상된다.