

# 최종 보고서

---

## 인터넷과웹기초

학 과	정보컴퓨터공학과
학 번	202025181
이 름	박소현
제 출 일	2022.06.04



## 내용

1. 기능 시나리오.....	3
기능 1. -----로그인 기능.....	3
기능 2. -----날씨에 따른 음식과 장소 추천 기능 .....	3
기능 3. -----검색창 기능과 검색결과를 지우는 기능 .....	3
2. 최소 개발 구현 .....	4
3. 추가 구현 [가산점].....	8

---

## 1. 기능 시나리오

---

각 기능 시나리오 별로 사용법을 **개괄식**으로 작성  
사용한 Open API 반드시 명시

---

### 기능 1. ----- 로그인 기능

기능1은 로그인 기능입니다. 흔히 생각할 수 있는 로그인 기능과는 달리 제가 만든 페이지의 로그인 기능은 1회성으로 사용되기 때문에 가입을 하지 않아도 사용을 할 수 있습니다.

로그인을 하면서 입력한 사용자의 정보는 localStorage에 저장되어 로그인 페이지를 나간 후에도 다른 페이지에서 활용할 수 있게 하였습니다.

여기서 Open API를 사용하였습니다. Open API는 OpenWeather라는 Weather API를 사용하였습니다.

사용자가 있는 지역을 입력값으로 받고 이를 활용해 다른 페이지에 해당 지역의 날씨가 나타납니다.

### 기능 2. ----- 날씨에 따른 음식과 장소 추천 기능

앞선 로그인 페이지를 벗어나면 Home으로 이동하게 됩니다. Home 페이지의 왼쪽 부분을 보게 되면 사용자가 입력한 정보와 날씨가 뜨게됩니다. Home에서는 페이지의 중앙부분에 추천 결과가 나오게 됩니다. FOOD 페이지와 PLACE 페이지에서는 앞선 정보와 추천 결과들이 왼쪽 부분에 나타나게 됩니다. 추천 결과는 제가 수집한 데이터를 가지고 날씨정보와 매치하여 날씨에 따라 결과값이 달라지게 설정하였습니다.

### 기능 3. ----- 검색창 기능과 검색 결과를 지우는 기능

FOOD 페이지와 PLACE 페이지로 가게 되면 검색창 기능을 사용할 수 있습니다. 검색어를 입력하게 되면 페이지의 중앙 부분에 검색결과가 나오게 됩니다. 그리고 이를 지우고 싶다면 오른쪽 하단에 remove 버튼을 누르게 되면 검색결과가 지워지게 됩니다. 검색창에 검색어를 입력하지 않고 검색을 하려고 한다면 검색어를 입력하라는 경고창이 나타나게 됩니다.

HOME에서는 검색창 기능을 사용할 수 없게 하였습니다. 따라서 검색어를 입력하고 검색을 누르거나 아무것도 입력하지 않은 상태에서 검색을 누르더라도 아무런 반응을 하지 않게 하였습니다.

여기서 Open API를 사용하였습니다. Open API는 카카오의 Daum 검색 API를 사용하였습니다.

---

## 2. 최소 개발 구현

---

최소 개발 기준을 어떻게 만족하였는지 **개괄식**으로 작성  
해당하는 소스코드 및 line number를 반드시 포함.

---

- **기능 시나리오 3가지 필수 구현**

- **로그인 기능**

1. 사용자의 이름을 입력 받는 부분 : start\_page.html의 #14
2. 사용자가 사는 지역을 입력 받는 부분 : start\_page.html의 #15-34
3. 로그인 버튼을 누르면 로그인 가능 여부를 판단하는 부분 : start.js의 #1-20

- **날씨에 따른 음식과 장소 추천 기능**

1. 사용자가 입력한 지역을 가지고 날씨를 반환하는 부분 : start.js의 #22-95
2. 날씨에 따른 음식과 장소를 추천하는 결과를 반환하는 부분 : start.js의 #97-151

- **검색창 기능과 검색결과를 지우는 기능**

1. 사용자가 검색어를 입력하지 않고 검색을 눌렀을 때 경고창을 띄우는 부분 :  
search\_button.js의 #1-7
2. 사용자가 검색어를 입력하고 버튼을 누르는 부분(실제로 검색기능이 작동되는 page  
만 작성하였습니다.) : FOOD.html의 #19-22, Place.html의 #19-22
3. 사용자가 입력한 검색어의 결과를 반환하는 기능을 하는 부분:  
FOOD.html의 #69-90, Place.html의 #69-99
4. 검색 결과를 지우는 부분: FOOD.html의 #29, #92-98, Place.html의 #29, #102-108

- **Open Web API 2개 이상 활용**

- OpenWeather API**

1. 해당지역에 따른 날씨 반환 하는 부분 : start.js의 #23-96

- 카카오의 Daum 검색 API**

1. 입력한 검색어의 결과를 반환하는 부분 : FOOD.html의 #69-90, Place.html의 #69-99

- 서로다른 <input> 3개 이상 활용

- 로그인 의 이름 입력 : start\_page.html의 #14
- 로그인 의 지역 입력 : start\_page.html의 #15-34
- 검색창의 검색어 입력 : FOOD.html의 #19-22, Place.html의 #19-22

- List 및 table 필수 활용

- Home 페이지에서 왼쪽부분에 홈페이지 사용법에 list 사용 : Homt.html의 #26-32
- PLACE 페이지에서 검색결과를 페이지에 표시할 때 table 사용 : Place.html의 #85-95

- Semantic (header, navigation, footer) 필수 활용

- 페이지 구획을 위해서 모든페이지에 Semantic 활용 : 모든 html 파일의 <body> tag내 부에 <header>, <aside>, <section>, <footer>를 사용 (start\_page.html에는 <aside>와 <footer>, intro.html에는 <aside> tag 사용 안함)

- Pseudo-class 2개 이상, pseudo-element 2개 이상, attribute selector 2개 이상, combinator 5개 이상 활용

- Pseudo-class는 :focus와 :hover를 사용. :focus는 로그인 시 사용자가 입력창의 내부 를 클릭하였을 때 입력창에 변화를 주기 위해 사용되었습니다. :hover는 사용자가 버튼 위에 커서를 올렸을 때 변화를 주기 위해 사용되었습니다.

1. :focus는 start.css의 #54-67

2. :hover는 intro.css의 #52, #124, home.css의 #52, #204, place.css의 #52, #162, food.css의 #52, #120

- pseudo-element는 ::before와 ::first-letter를 사용. ::before는 검색 결과 값을 창에 표 시 할 때 결과 값 중 링크가 있는 결과값에 대한 표시를 해주기 위해 사용되었습니 다. ::first-letter는 검색 결과 값을 창에 표시 할 때 첫번째 글자에 강조를 주기 위해 사 용되었습니다.

1. ::before는 food.css의 #180

2. ::first-letter는 food.css의 #191

- attribute selector는 start.css와 place.css에 사용 : start.css의 #15 , place.css의 #12

- combinator는 모든 css에서 볼 수 있다. : start.css, intro.css, home.css, food.css, place.css

- **Float 및 clear 활용하여 웹의 layout 구성 필수**

- Home, FOOD, PLACE 페이지에서 사용자의 정보 표시하는 곳에 사용

- 1. home.css의 #96, #117, #128, #136, #147, #171, #267, #279, #286

- 2. food.css의 #96, #125, #136, #144, #153, #162, #210, #222, #229, #238, #247

- 3. place.css의 #96, #108, #119, #127, #136, #145, #198, #210, #217, #226, #235

- **모바일 환경 대응 구현 필수 (width: 900px 이하)**

- Home, FOOD, PLACE 페이지에서 @media only screen and (max-width: 900px) 사용 : home.css의 #219-288, food.css의 #207-249, place.css의 #195-237

- 모든 css에서 margin, width, font-size처럼 크기가 들어가는 요소는 px로 고정시키는 것이 아닌 대부분 %로 유동적으로 변경될 수 있게 함

- **flex 혹은 grid 한번 이상 필수 활용**

- 모든 css에서 flex를 사용

- 1. start.css의 #8, #19, #38, intro.css와 home.css와 food.css, 그리고 place.css의 #22, #36

- **DOM 활용하여 HTML element 혹은 CSS property 합쳐서 5개 이상 다루기**

- 입력창에 입력한 정보를 가져오기 위해 search\_button.js와 start.js에서 활용

- 1. search\_button.js의 #1, start.js의 #1-2, #24

- Home.html, FOOD.html, Place.html에서 활용. Home.html, FOOD.html 그리고 Place.html에서는 사용자의 정보와 날씨를 표시하기 위해 사용하였습니다. Home.html에서는 추천결과를 화면에 표시하기 위해 사용되었습니다.

- 1. Home.html의 #42-50, #66-69, #83-86

- 2. FOOD.html의 #35-43, ##48-51, #56-59

- 3. Place.html의 #35-43, ##48-51, #56-59

- **입력 Format, 네트워크 연결 등의 에러 처리 기능 필수 구현**

- 로그인 할 때 모두 입력하지 않고 join버튼을 눌렀을 때 경고창 뜨게 하기

- 1. start.js의 #6-7

- 검색어 창에 검색어를 입력하지 않고 검색어를 입력하려고 했을 때 경고창 뜨게 하기

1. search\_button.js의 #4-6

- 수업시간에 배운 jQuery/AJAX/JSON 기능 활용

- 검색결과를 가져오기 위해 jQuery와 AJAX를 FOOD.html과 PLACE.html에서 사용

1. FOOD.html의 #69-97, PLACE.html의 #69-107

- 날씨 값과 그에 따른 추천결과를 가져오기 위해 JSON을 start.js과 Home.html, FOOD.html, PLACE.html에서 사용

1. start.js의 #81-95, Home.html의 #64, #81, FOOD.html과 Place.html의 #46, #54

---

### 3. 추가 구현 [가산점]

---

최소 개발 기준 이외에 추가 구현한 부분을 **개괄식**으로 작성  
해당하는 소스코드 및 line number를 반드시 포함.

---

- 사용자가 입력한 정보가 유지가 된채로 페이지가 변경되어도 계속해서 남을 수 있도록 함.

- start.js의 #13-14

- 사용자가 입력한 정보를 토대로 Open API를 활용하였을 때, Open API에게서 받은 정보를 몇가지의 경우로 구분하여 결과값을 달리 도출함. 이는 사용자가 같은 지역을 입력하더라도 OpenWeather라는 Open API에서 받은 날씨의 종류에 따라 추천결과가 달라지게함. 이 역시도 날씨의 정보와 추천결과가 유지된 채로 페이지가 변경되어도 계속해서 남을 수 있도록 함.

- start.js의 #93-150