

최종 보고서

인터넷과웹기초

학 과	정보컴퓨터공학전공
학 번	201924648
이 름	나인수
제 출 일	2023-06-07



부산대학교
PUSAN NATIONAL UNIVERSITY

내용

1. 기능 시나리오.....	3
기능 1. ----	3
기능 2. ----	3
기능 3. ----	3
2. 최소 개발 구현	4
3. 추가 구현 [가산점].....	13

제출 전에 목차 전체 업데이트 필수

1. 기능 시나리오

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

기능 1. Chat-GPT 사용하기

- Open AI 사이트에서 REST API 형태로 제공해주는 기능을 사용하면 웹 브라우저에 나만의 chat-bot을 만들 수 있다.
- 기본적인 UI를 제공해주면서 원활하게 chat-bot과 상호작용을 할 수 있다.
- 기능 3을 위해서 채팅마다 기록한다.

기능 2. DALL-E2 사용하기

- 이미지를 생성해주는 모델을 Open AI 사이트에서 REST API를 사용하면 웹 브라우저에 서비스할 수 있다.
- 한번의 prompt를 입력 시 4개 또는 8개의 이미지를 생성해서 보여준다.
- 이미지를 출력할 때 일정한 간격으로 정렬해서 보기 좋게 출력한다.

기능 3. 채팅 로그 보여주기

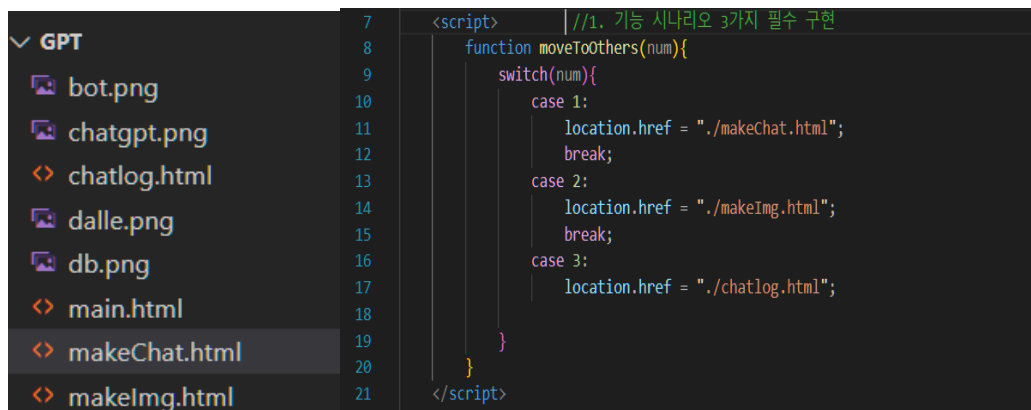
- 자바스크립트는 보안 문제로 인해서 파일 입출력 같은 기능을 사용하기 어렵다.
- 하지만 Web Storage API 를 이용한다면 JSON 형태로 브라우저에 데이터를 영구적으로 저장할 수 있다.
- 이를 통해서 채팅 로그를 기록하고 TABLE 형태로 출력해주는 기능을 구현한다.

2. 최소 개발 구현

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

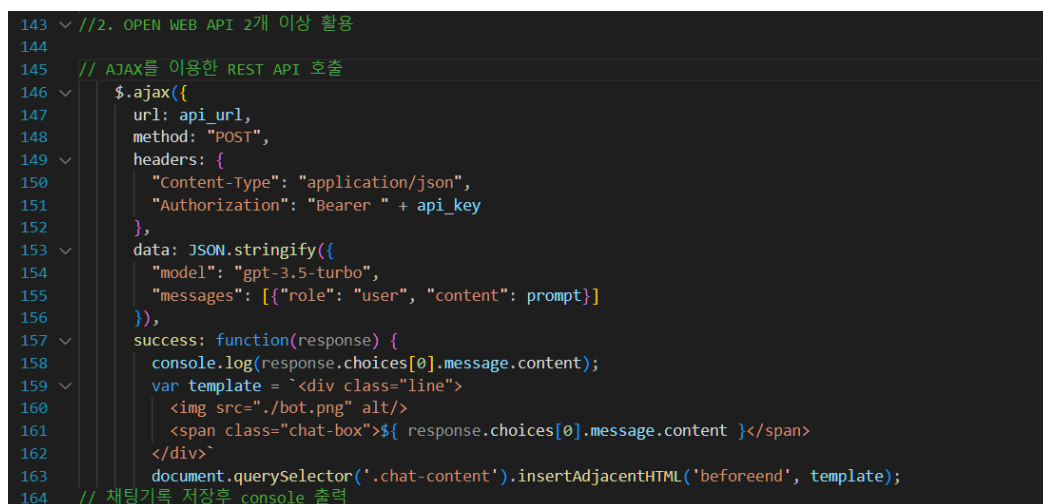
1. 기능 시나리오 3가지 필수 구현

서로 다른 html 파일 3개를 만들고 main.html에서 인덱싱 하는 것으로 하였다. 코드 넘버는 main.html의 7번 째 줄이다.



2. Open Web API 2 개 이상 활용

첫 번째로는 makeChat.html 의 143 번 째 줄에 있는 코드이다. 이는 chat GPT 의 기능을 가져올 때 사용한다.



두 번째로는 이미지 생성 모델을 가져오는 API 이다. 이는 makelmg.html 의 81 번째 줄부터 시작한다.

```

79
80 // 2.Open Web API 2개 이상 활용
81
82 // AJAX를 이용한 REST API 호출
83 $.ajax({
84     url: api_url,
85     method: "POST",
86     headers: {
87         "Content-Type": "application/json",
88         "Authorization": "Bearer " + api_key
89     },
90     data: JSON.stringify({
91         "prompt": myPrompt,
92         "n": n,
93         "size": "1024x1024"
94     }),
95     success: function(response) {
96         document.querySelector('#input').innerHTML="";
97         var title = document.createElement("p");
98         title.innerHTML = "prompt : "+myPrompt;
99
100         var date = document.createElement("p");
101         var currentDate = new Date();
102         var dateString = currentDate.toLocaleDateString();
103         var timeString = currentDate.toLocaleTimeString();
104         var dateTimeString = dateString + " " + timeString;
105         date.innerHTML = dateTimeString;
106
107         var container = document.createElement("div");
108         container.className = "image-container";
109         var imageUrls = response.data;
110

```

3. 서로 다른 <input> 3 개 이상 활용

```

50 <!--3. 서로 다른 <input> 3개 이상 활용 -->
51 <section id="id2">
52     <form style="display: inline-block; margin-right: 30px;">
53         <label>
54             4개
55         <input type="radio" name="option" checked="checked"/>
56         </label>
57         <label>
58             8개
59         <input type="radio" name="option" />
60         </label>
61     </form>
62     <input class="chat-box" id="input" value="an astronaut playing basketball with cats in space, digital art">
63     <button id="send">전송</button>
64 </section>

```

makeImg.html 파일에서 50 번째 줄에서 radio, button, text 이렇게 3 가지를 활용했다.

4. List 및 table 필수 활용

```

85 <table id="myTable">
86     <thead>
87         <tr>
88             <th>날짜</th>
89             <th>역할</th>
90             <th>내용</th>
91         </tr>
92     </thead>
93     <tbody>
94     </tbody>
95 </table>

```

chatlog.html 에서 85 줄에 있는 코드이다. Table 을 미리 정의하고 이를 자바스크립트에서 row 를 추가한다.

5. Semantic (header, navigation, footer) 필수 활용

```
65      <!--5. Semantic (header, navigation, footer) 필수 활용-->
66      <header>
67      |   <h1>인공지능 활용 서비스</h1>
68      </header>
69      <navigation>
70      <form class="func">
71      |   <button type="button" onclick='moveToOthers(1);'>
72      |   |   <div class="cell">
73      |   |   |   
74      |   |   </div>
75      |   |   <p>채팅봇 이용하기</p>
76      |   </button>
77      </form>
78
79      <form class="func">
80      |   <button type="button" onclick='moveToOthers(2);'>
81      |   |   <div class="cell">
82      |   |   |   
83      |   |   </div>
84      |   |   <p>이미지 생성하기</p>
85      |   </button>
86      </form>
87
88
89      <form class="func">
90      |   <button type="button" onclick='moveToOthers(3);'>
91      |   |   <div class="cell">
92      |   |   |   
93      |   |   </div>
94      |   |   <p>채팅 로그확인하기</p>
95      |   </button>
96      </form>
97      </navigation>
98
99      <footer>
100     |   <p>copyright &copy; 201924648 nainsoo</p>
```

Main.html 코드이다. 64 번 코드부터 시작하며 64 번 줄에 <header> 69 번 줄에 <navigation> 99 번 줄에 <footer>가 있다.

6. Pseudo-class 2 개 이상, pseudo-element 2 개 이상, attribute selector 2 개 이상,

combinator 5 개 이상

Pseudo-class 2 개 이상은 main.html 의 34 번 줄에서 구현됐다.

```
34      /* 6. Pseudo-class 2개 이상, pseudo-element 2개 이상, attribute selector 2개 이상, combinator 5개 이상 */
35      .func{
36          margin: 10px;
37      }
38
39      .cell{
40          margin: 10px;
41          align-items: center;
42      }
43
```

pseudo-element 2 개 이상 은 makeImg.html 8 번 줄에서 시작한다.

```
8      /* 6. Pseudo-class 2개 이상, pseudo-element 2개 이상, attribute selector 2개 이상, combinator 5개 이상 */
9      #id1{
10          text-align: center;
11          background-color: #bisque;
12          font-style: bold;
13          border-radius: 30px;
14      }
15      #id2{
16          text-align: center;
17      }
```

attribute selector 2 개 이상 은 makeChat.html 파일 24 번 줄에서 시작한다.

```
24      /* 6. Pseudo-class 2개 이상, pseudo-element 2개 이상, attribute selector 2개 이상, combinator 5개 이상 */
25
26      section[class~="box"]{
27          text-align: center;
28      }
29
30      header[class="box"]{
31          text-align: center;
32      }
```

combinator 5 개 이상은

chatlog.html 파일에서 2 개

```
22      /* 6. Pseudo-class 2개 이상, pseudo-element 2개 이상, attribute selector 2개 이상, combinator 5개 이상 */
23
24      header h1{
25          text-align: center;
26          background-color: #bisque;
27          font-style: bold;
28          border-radius: 30px;
29      }
30
31      table {
32          border: 2px solid;
33          border-collapse: collapse;
34          text-align: center;
35          width: 100%;
36      }
37
38      th, td{
39          border: 1px solid;
40          padding: 10px;
41      }
```

Main.html 에서 3 개가 있다.

```

29      /* 6.   Pseudo-class 2개 이상, pseudo-element 2개 이상, attribute selector 2개 이상, combinator 5개 이상 */
30
31      .cell img {
32          width: 250px;
33          height: 250px;
34          object-fit: cover;
35          border-radius: 30px;
36      }
37

```

```

52      /* 6.   Pseudo-class 2개 이상, pseudo-element 2개 이상, attribute selector 2개 이상, combinator 5개 이상 */
53      header h1{
54          background-color: #bisque;
55          font-style: bold;
56          border-radius: 30px;
57      }
58
59      .func button{
60          background-color: #bisque;
61          border-radius: 30px;
62      }
63

```

7. Float 및 clear 활용하여 웹의 layout 구성 필수

```

40      /* 7.   Float 및 clear 활용하여 웹의 layout 구성 필수 */
41      nav{
42          border: 1px solid #FF1493;
43          width: 150px;
44          float: left;
45          margin: 10px;
46          margin-top: 0px;
47          padding: 5px;
48          background-color: gray;
49      }
50

```

makeChat.html 의 40 번줄에서 nav 에 float 을 넣었다. nav 에는 다른 페이지로 가는 a 태그가 존재한다. 기존 채팅 결과 표시를 해주는 UI 는 section 에 있고 nav 에 float 을 띄워서 옆에 보이게 설정했다. 결과는 다음 화면과 같다.

채팅봇 이용하기

채팅하기
로그보기
이미지만들기

전송

8. 잘못된 동작에 대한 에러 핸들링 두개 이상 구현 필수

makeChat.html 파일의 124 번 줄에서 prompt 를 전송하는 버튼을 눌렀을 때 값이 없으면 바로 return 을 하게 만들었다. 마찬가지로

```
124 // 8. 잘못된 동작에 대한 에러 핸들링 두개 이상 구현 필수
125 if(document.querySelector('#input').value==""){
126     return;
127 }
128
```

makeImg.html 파일의 69 번 줄에서도 input 값이 없으면 전송하는 버튼을 눌러도 아무것도 하지 않게 바꿨다.

```
69 // 8. 잘못된 동작에 대한 에러 핸들링 두개 이상 구현 필수
70 if(document.querySelector('#input').value==""){
71     return;
72 }
73
```

9. 모바일 환경 대응 구현 필수 (width: 900px 이하일때 동작, 기존 웹사이트는 width: 1060px 로 구현)

딱히 width 가 줄어든다고 해서 문제가 되는 건 아니었기에 모바일 환경을 인식하고 동작한다는 것만 보여주기 위해서 가장 간단한 페이지에서 배경색상을 반전하는 기능만 구현했다. 그래서 chatlog.html 의 8 번 줄에 구현했다.

채팅 로그 확인

날짜	역할	내용
2023. 6. 4. 오후 9:36:24	me	안녕하세요.
2023. 6. 4. 오후 9:36:24	bot	안녕하세요! 저는 AI 어시스턴트입니다. 무엇을 도와드릴까요?
2023. 6. 4. 오후 9:37:02	me	너 누구야?
2023. 6. 4. 오후 9:37:02	bot	저는 AI 어시스턴트로, 이용자의 질문에 대답하고 작업을 수행하는 봇입니다. 어떤 도움이 필요하신가요?
2023. 6. 4. 오후 9:55:39	me	자바스크립트에 대해서 알려줘

채팅 로그 확인

날짜	역할	내용
2023. 6. 4. 오후 9:36:24	me	안녕하세요.
2023. 6. 4. 오후 9:36:24	bot	안녕하세요! 저는 AI 어시스턴트입니다. 무엇을 도와드릴까요?
2023. 6. 4. 오후 9:37:02	me	너 누구야?
2023. 6. 4. 오후 9:37:02	bot	저는 AI 어시스턴트로, 이용자의 질문에 대답하고 작업을 수행하는 봇입니다. 어떤 도움이 필요하신가요?

```
8 /* 9. 모바일 환경 대응 구현 필수 (width: 900px 이하일때 동작, 기존 웹사이트는 width: 1060px로 구현)*/
9 @media screen and (min-width: 1060) {
10     body {
11         background-color: white;
12         color: black;
13     }
14 }
15
16 @media screen and (max-width: 900px) {
17     body {
18         background-color: black;
19         color: white;
20     }
21 }
```

10. flex 혹은 grid 한번 이상 필수 활용

```

23      /*10. flex 혹은 grid 한번 이상 필수 활용*/
24      navigation{
25          display: flex;
26          flex-wrap: wrap;
27          justify-content: center;
28      }

```

Main.html 파일에서 23 번 줄 에 flex 를 사용했다. 이를 통해서 아래 보이는 것처럼 form 태그들을 보기 좋게 정렬 해서 출력했다.

```

76      <navigation>
77      <form class="func">
78          <button type="button" onclick='moveToOthers(1);'>
79              <div class="cell">
80                  
81              </div>
82              <p>채팅봇 이용하기</p>
83          </button>
84      </form>
85
86      <form class="func">
87          <button type="button" onclick='moveToOthers(2);'>
88              <div class="cell">
89                  
90              </div>
91              <p>이미지 생성하기</p>
92          </button>
93      </form>
94
95
96      <form class="func">
97          <button type="button" onclick='moveToOthers(3);'>
98              <div class="cell">
99                  
100             </div>
101             <p>채팅 로그확인하기</p>
102          </button>
103      </form>
104  </navigation>

```



11. DOM 활용하여 HTML element 혹은 CSS property 합쳐서 5 개 이상 다루기

아래 코드는 makelmg.html 의 101 번 줄이다. 이는 REST API 를 통해서 이미지를 4 개 또는 8 개를 받아오면 URL 을 여러 개를 얻게 된다. 이를 정렬해서 태그를 통해 출력하기 위해서는 DOM 을 활용해서 동적으로 HTML element 를 여러 개를 만들어야 한다. 밑에 코드는 적어도 5 개이상 DOM 을 활용한다.

```

101 //11. DOM 활용하여 HTML element 혹은 CSS property 업저서 5개 이상 다루기
102 document.querySelector('#input').innerHTML="";
103 var title = document.createElement("p");
104 title.innerHTML = "prompt : "+myPrompt;
105
106 var date = document.createElement("p");
107 var currentDate = new Date();
108 var dateString = currentDate.toLocaleDateString();
109 var timeString = currentDate.toLocaleTimeString();
110 var dateTimeString = dateString + " " + timeString;
111 date.innerHTML = dateTimeString;
112
113 var container = document.createElement("div");
114 container.className = "image-container";
115 var imageUrls = response.data;
116
117 imageUrls.forEach(function(url) {
118     var cell = document.createElement("div");
119     cell.className = "image-cell";
120
121     var img = document.createElement("img");
122     var strUrl = url.url;
123     img.src = strUrl;
124
125     cell.appendChild(img);
126     container.appendChild(cell);
127     document.querySelector(".chat-content").appendChild(date);
128     document.querySelector(".chat-content").appendChild(title);
129     document.querySelector(".chat-content").appendChild(container);
130 });

```

12. 입력 Format, 네트워크 연결 등의 에러 처리 기능 필수 구현

```

132 //12. 입력 Format, 네트워크 연결 등의 에러 처리 기능 필수 구현
133 error: function(error) {
134     console.log(error);
135 }
136 });

```

REST API 를 이용하면서 error 가 발생시 console 을 통해서 에러를 알려주는 기능을 추가했다. makelmg.html 파일의 132 번 줄이다.

13. 1

REST API 를 호출할 때 AJAX 를 사용했다. 밑의 코드는 makeChat.html 파일의 143 번 줄이다.

```

147 //2. OPEN WEB API 2개 이상 활용
148 //13. 수업시간에 배운 jQuery/AJAX/JSON 기능 활용
149 // AJAX를 이용한 REST API 호출
150 $.ajax({
151     url: api_url,
152     method: "POST",
153     headers: {
154         "Content-Type": "application/json",
155         "Authorization": "Bearer " + api_key
156     },
157     data: JSON.stringify({
158         "model": "gpt-3.5-turbo",
159         "messages": [{"role": "user", "content": prompt}]
160     }),
161     success: function(response) {
162         console.log(response.choices[0].message.content);
163         var template = `<div class="line">
164             
165             <span class="chat-box">${ response.choices[0].message.content }</span>
166         </div>`;
167         document.querySelector('.chat-content').insertAdjacentHTML('beforeend', template);
168         // 채팅기록 저장후 console 출력
169         data = {
170             "date" : currDate,
171             "job" : "bot",
172             "content" : response.choices[0].message.content
173         };

```

14. 소스 코드에 각 함수 및 최소 개발 기준 별로 기능 설명하는 주석 필수

3. 추가 구현 [가산점]

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

Web Storage API 을 이용해서 웹 브라우저에 json형식의 파일을 저장하는 기능을 구현했다.

- 데이터를 읽는 코드는 chatlog.html 의 46번 줄이다.
- 저장하는 코드는 makeChat.html의 63번 줄에 있다.

```

46 <script>
47     function read(name){
48         // 저장된 JSON 데이터 가져오기
49         var storedData = localStorage.getItem(name);
50
51         // JSON 문자열을 객체로 변환
52         var parsedData = JSON.parse(storedData);
53
54         //table element 가져오기
55         var table = document.getElementById("myTable");
56
57         parsedData.arr.forEach(element => {
58             var newRow = table.insertRow();
59             // 가져온 데이터 사용
60             var one = newRow.insertCell(0);
61             var two = newRow.insertCell(1);
62             var three = newRow.insertCell(2);
63
64             one.innerText = element.date;
65             two.innerText = element.job;
66             three.innerText = element.content;
67         });
68

```

```

63  ▾ function store(name, data){
64      var storedData = localStorage.getItem(name);
65
66  ▾      if (storedData) {
67          var parsedData = JSON.parse(storedData);
68          parsedData.arr.push(data);
69          localStorage.setItem(name, JSON.stringify(parsedData));
70      }else{
71  ▾          var temp = {
72              "arr" : []
73          }
74          temp.arr.push(data);
75          localStorage.setItem(name, JSON.stringify(temp));
76      }
77  }
78

```