

Hyun's Note

2 0 1 8 2 4 5 0 9

송 현 아

인 터 넷 과 웹 기 초 텀 프 로젝 트

AN OVERVIEW

" 단순히 기록만 하는 것이 아닌 그 이상의
필요한 기능들을 갖춘 나만의 노트 애플리
케이션 입니다. "

01

To-Do List

02

Search

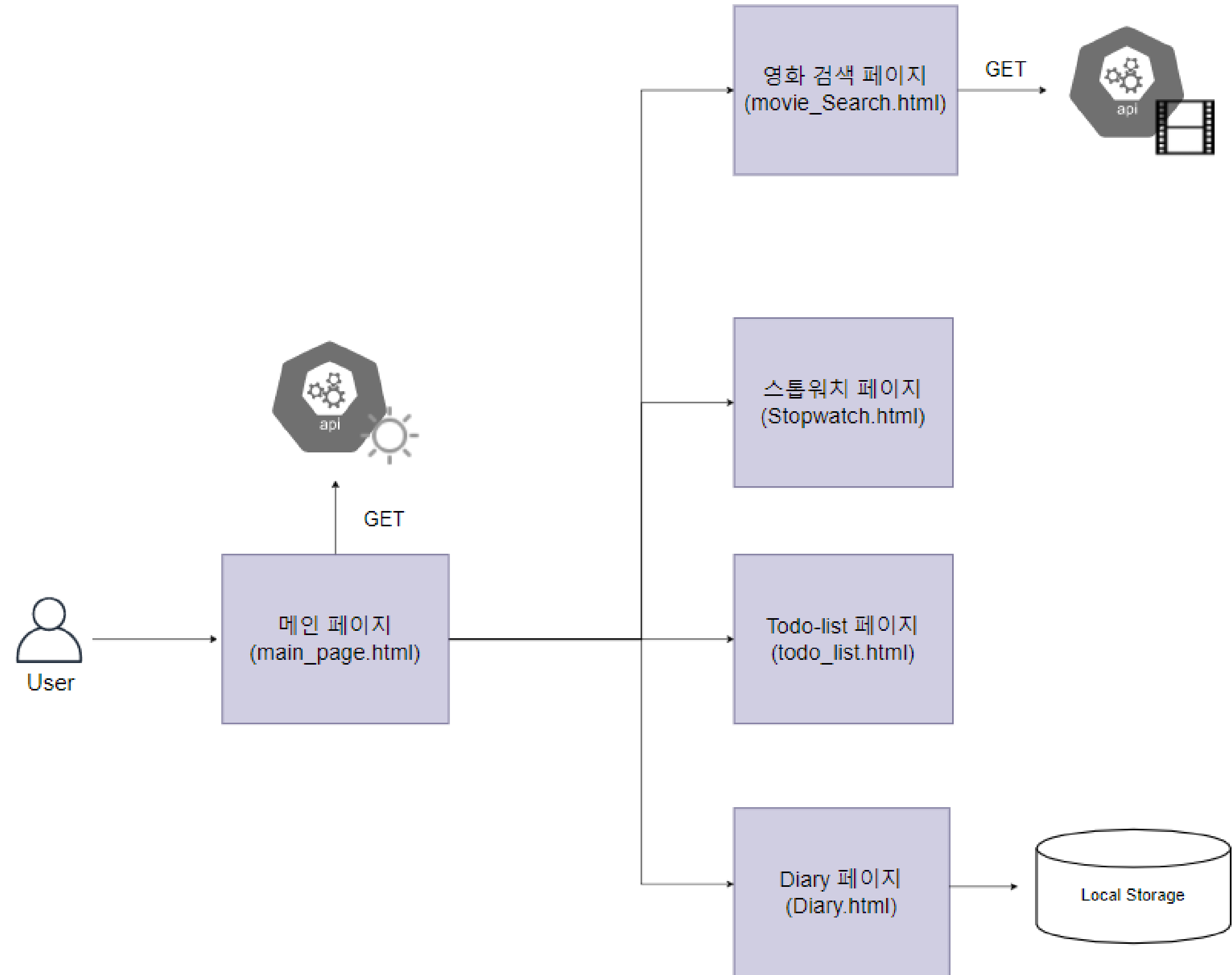
03

Stopwatch

04

Diary

웹 페이지 구성도



01.

최소 개발 기준

- OPEN API 2개 이상 활용
- 서로 다른 HTML element 15개 이상
- 서로 다른 CSS selector로 15 개 이상 스타일 사용
- JavaScript를 통해 3가지 이상의 동적 기능 제공
- JavaScript를 활용해 부적절한 입력 및 OPEN API의 response 중 에러에 대한 처리
- jQuery/AJAX/JSON 기능 활용
- Node.js를 활용한 local webserver를 사용하는 경우, web server의 기능은 routing 과 API server로부터 받은 data를 forwarding하는 역할만 구현 할 것

Open API

1. 실시간 날씨 조회 API

: 실시간 날씨 조회를 통해서 현재 온도와 습도를 출력

- URL : <https://openweathermap.org/current>

.- apiURL : [http://api.openweathermap.org/data/2.5/weather?q={city name}&appid=\[{API key}\]](http://api.openweathermap.org/data/2.5/weather?q={city name}&appid=[{API key}])

```
const getJSON = function(url,callback){
  const xhr = new XMLHttpRequest();
  xhr.open('GET',url,true);
  xhr.responseType = 'json';
  xhr.onload = function(){
    //접속이 성공적이면 null 값 반환, 그 외에는 status 값 반환
    const status = xhr.status;
    if(status === 200){
      callback(null, xhr.response);
    }else{
      callback(status,xhr.response);
    }
  };
  xhr.send();
};

getJSON('https://api.openweathermap.org/data/2.5/weather?q=busan&appid=8a2d42a5944d266c120da612439b6983&units=metric',
function(err,data){
  // null 값이 아니면 err
  if(err !== null){
    alert('죄송합니다. 예상치 못한 오류가 발생했습니다.' + err);
  }else{
    document.getElementById("weather").innerHTML = "Today's weather in " + data.name+" is "+ data.weather[0].main + "<br> Temperature : "+data.main.temp+"<br> Humidity : "
    +data.main.humidity;
  }
});
</script>
```

OPEN API의 response 중 에러에 대한 처리

Open API

2. 네이버 영화 검색 API “Node.js 활용”

: 입력받은 영화 제목, 키워드, 장르 등을 통해서 해당 영화 정보를 출력

- URL : <https://developers.naver.com/docs/search/movie/>
- apiURL : <https://openapi.naver.com/v1/search/movie.json>

```
function displayResponse(){
    if(document.getElementById('query').value.length > 30)
    {
        alert("query length error");
        return;
    }
    // 로컬 호스트에 요청을 보냄
    var url = "http://localhost:3000/movie";
    var query = document.getElementById('query').value;
    var genre = document.getElementById('genre').value;
    url = url + "?" + "query="+query + " &genre="+genre
    xhr.open("GET", url);
    xhr.send();
}
```

</script>

HTML Element

```
<!-- 본문 작성 -->  
<body>  
    <header>  
        <h1>Hyun's Page</h1>  
        <p>Do Your Best!</p>  
    </header>  
    <nav>  
        <a href="movie_search.html">[ Movie ]</a>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~  
        <a href="todo_list.html">[ To Do List ]</a>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~  
        <a href="add_schedule.html">[ Add Diary ]</a>&nbsp;&nbsp;&nbsp;&nbsp;&~  
        <a href="Diary.html">[ Diary ]</a>&nbsp;&nbsp;&nbsp;&~  
        <a href="StopWatch/stopwatch.html">[ StopWatch ]</a>&nbsp;&nbsp;&nbsp;&~  
        <form method=get action="http://www.youtube.com/results" target="_blank" >  
            YouTube  
            <input type=text name=q size=20% maxlength=255 value="" /> <!-- 유튜브 검색 입력 창 -->  
            <input type=submit name=btnG2 value="Search" /> <!-- 검색 버튼 -->  
            <p id="weather"></p>  
        </form>  
    </nav>  
    <section id="google">  
        <form method=get action="http://www.google.co.kr/search" target="_blank" >  
            <input type=text name=q size=100% maxlength=255 value="" /> <!-- 구글 검색 입력 창 -->  
            <input type=submit name=btnG value="Search" /> <!-- 검색 버튼 -->  
        </form>  
    </section>  
</script>
```

- <h1>, <p>
- <header>, <section>, <nav>
- <a>
- <form>, <input>

```
</head>
<body>
  <div style="text-align: center">
    <h1 id="now" style="color: white"></h1>
  </div>
  <!-- 타이머 출력하는 부분 -->
  <div id="timer">
    00:00:00
  </div>
  <!-- 타이머 컨트롤 버튼 -->
  <div id="buttons">
    <button id="start_btn" onclick="startTimer();">Study Start</button>
    <button id="pause_btn" onclick="stopTimer();">Study Stop</button>
    <button id="reset_btn" onclick="resetTimer();">Time Reset</button>
  </div>
  <ul id="myUL"></ul>
  <script src="studyTimer.js"></script>
</body>
</html>
```

- `<div>`
- `<button>`
- ``, ``

```
<form name="myform" method="get" action="#">
<!-- 일정 제목 입력 -->
<div class="box1">
| <h1>Add Diary :)</h1>
</div>
<div class="box2">
<div>
| <!-- 일정 입력 -->
| <label for="fname">Title</label><br>
| <input type="text" name="fname2" id="fname" placeholder="제목 입력">
</div>
<br>
<div>
| <!-- 일기 날짜 입력 -->
| <label for="date">Date</label><br>
| <span><input type="date" id="date"></span>
</div>
<br>
<div>
| <!-- 일정 카테고리 -->
| <label for="interest">Emotion</label><br>
| <div>
행복<input type="radio" name="category" value="happy" >&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~
슬픔<input type="radio" name="category" value="sad" >&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~
차분함<input type="radio" name="category" value="calm" >&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~
화남<input type="radio" name="category" value="angry">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~
놀람<input type="radio" name="category" value="surprised" >&nbsp;&nbsp;&nbsp;&nbsp;&~
</div>
</div>
<br>
<div>
| <!-- 메모 -->
| <label for="memo">Content</label><br>
| <textarea id = "memo" name="etc" cols="50" rows="4"></textarea>
</div>
```

- `<label>`
- `
`
- `<textarea>`
- ``

CSS Selector

- .box1 /.box2 : Selects all elements with class
- div>label
- div>div>div

```
<style>
.box2{
  padding: 10px;
  margin: 1%;
  /* width: 90% */
}
.box1{
  border: 2px solid #b1b6e3;
  background-color: #b1b6e3;
  color: white;
  text-align: center;
  padding-bottom: 20px;
}
div>label{
  display: inline-block;
  background-color: white;
  width: 100%;
  height: 25px;
  font-size: 14px;
}

div>div>div{
  border: 1px solid #8c8c8c;
  padding-left: 40%;
  padding-bottom: 10px;
  padding-top: 5px;
}
```

```
nav input{
  border: 1px solid #b1b6e3;
  padding: 4px 10px;
}
/* 마우스를 올릴때 안올릴 때 */
nav a:link, nav a:visited{
  color: white;
}
/* 마우스를 올렸을 때 배경 색 변경 */
nav a:hover{
  background-color: white;
  color: #8386a2
}
/* -----메인 콘텐츠----- */
/* section design */
section{
  margin-top: 30px;
  margin-left: 30%;
  width:90%;
  height:200px;
}
/* -----검색창----- */
section input[type=submit]{
  border-radius: 15px;
  margin: 10px 0;
  padding: 10px 12px;
  border: 1px solid gray;
}
section input[type=text]{
  border-radius: 15px;
  margin: 10px 0;
  padding: 10px 12px;
  border: 1px solid #b1b6e3;
}
```

- # date
- #result

- nav input
- nav a:link, nav a:visited
- nav a:hover
- section input[type=submit]

```
input[type=button],input[type=reset]{
  background-color: #b1b6e3;
  width: 49.5%;
  border: none;
  padding: 10px;
  /* margin-left: 3%; */
  /* margin: 7px; */
  color: white;
  box-sizing: border-box;
  margin-top: 30px;
}
textarea{
  width: 100%;
}

#date{
  border: 1px solid #8c8c8c;
  display: inline-block;
  margin: 4px;
  height: 25px;
  font-size: 12px;
  width: 170px;
}
button{
  float: right;
  border-radius: 15px;
  margin: 10px 5px;
  padding: 10px 12px;
  border: 1px solid gray;
}
#result{
  margin: 30px;
}
```

```
ul li {
  cursor: pointer;
  position: relative;
  padding: 12px 8px 12px 40px;
  list-style-type: none;
  background: #eee;
  font-size: 18px;
  transition: 0.2s;
}
ul li:nth-child(odd) {
  background: #f9f9f9;
}
ul li:hover {
  background: #ddd;
}
ul li.checked {
  background: #888;
  color: #fff;
  text-decoration: line-through;
}

/* Add a "checked" mark */
ul li.checked::before {
  content: '';
  position: absolute;
  border-color: #fff;
  border-style: solid;
  border-width: 0 2px 2px 0;
  top: 10px;
  left: 16px;
  transform: rotate(45deg);
}
```

- ul li
- ul li:nth-child(odd)
- ul li:hover
- ul li.checked::before

JavaScript 동적 기능(1)

```
</div>
<!-- 타이머 컨트롤 버튼 -->
<div id="buttons">
  <button id="start_btn" onclick="startTimer();">Study Start</button>
  <button id="pause_btn" onclick="stopTimer();">Study Stop</button>
  <button id="reset_btn" onclick="resetTimer();">Time Reset</button>
</div>
<ul id="myUL"></ul>
```

→ 버튼 입력에 따라 해당 함수 실행함

```
// StudyStart 버튼을 누르는 경우 => 시간이 지속적으로 흐름
function startTimer() {
  if (stoptime == true) {
    stoptime = false;
    timerCycle();
  }
}
```

```
// StudyStop 버튼을 누르는 경우 => 시간이 일시 정지
function stopTimer() {
  if (stoptime == false) {
    stoptime = true;
    var li = document.createElement("li");
    var t = document.createTextNode(result_time);
    li.appendChild(t);
    document.getElementById("myUL").appendChild(li);
    var span = document.createElement("span");
    var txt = document.createTextNode("\u00D7");
    span.className = "close";
    span.appendChild(txt);
    li.appendChild(span);

    for (i = 0; i < close.length; i++) {
      close[i].onclick = function() {
        var div = this.parentElement;
        div.style.display = "none";
      }
    }
  }
}
```

```
// Time Reset 버튼을 누르는 경우 => 시간 리셋 함수
function resetTimer() {
  time.innerHTML = '00:00:00';
  min = 0;
  sec = 0;
  hour = 0;
}
```

JavaScript 동적 기능(2)

Add Diary :)

Title : 오늘은 나의 생일

Date : 2021-06-02

Emotion

행복 ☒ 슬픔 ☐ 기쁨 ☐ 화남 ☐ 놀람 ☐

Content

오늘은 드디어 나의 생일이다!! 맛있는 것도 마니 먹고 무지 행복한 날이었다!!

비우기

취소

저장

→ 저장 버튼을 누르면 Form에 입력한
일기 내용에 따라 show()를 통해서
화면에 출력함

```
<script>

// Create an object:
function show(){
  for (var i=0; i<5; i++) {
    if (document.getElementsByName("category")[i].checked == true) {
      e = document.getElementsByName("category")[i].value;
    }
  }
  //JSON data
  var memo = {
    title: document.getElementById('fname').value,
    date: document.getElementById('date').value,
    emotion: e,
    content:document.getElementById('memo').value
  };

  localStorage.setItem(memo["date"],JSON.stringify(memo));
  // 결과 출력
  document.getElementById("result").innerHTML += "<h1> 제목: " + memo["title"] + "</h1>" + "날짜: " + memo["date"] + "<br>" + "기분: " + memo["emotion"] + "<br><br>" + memo["content"] ;
  document.getElementById("result").style.border = "1px solid #999";
}
```

→ 비우기 버튼을 누르면 화면에 출력되었던
일기 내용을 사라지게 함

```
}
function bin(){
  document.getElementById("result").innerHTML = "" ;
  document.getElementById("result").style.border = "none";
}
```

JavaScript 동적 기능(3)

To Do List

일정 추가

```
}, false);
// 일정 추가하면 리스트 추가됨
function newElement() {
    var li = document.createElement("li");
    var inputValue = document.getElementById("myInput").value;
    var t = document.createTextNode(inputValue);
    li.appendChild(t);
    if (inputValue === '') {
        alert("You must write something!");
    } else {
        document.getElementById("myUL").appendChild(li);
    }
    document.getElementById("myInput").value = "";

    var span = document.createElement("span");
    var txt = document.createTextNode("\u00D7");
    span.className = "close";
    span.appendChild(txt);
    li.appendChild(span);

    for (i = 0; i < close.length; i++) {
        close[i].onclick = function() {
            var div = this.parentElement;
            div.style.display = "none";
        }
    }
}
</script>
</body>
```

→ 입력란에 일정을 입력하고 확인 버튼을 누르면 newElement() 함수가 실행되면서 새로운 리스트 목록이 추가됨

→ 새로 생성된 리스트에 close 기능이 포함됨

빈 문자열로 입력하는 경우에는 alert를 이용해서 에러에 대한 처리

jQuery/AJAX/JSON 기능 활용

Movie List

Find your movie

Movie Genre							
1	2	4	7	11	12	15	16
드라마	판타지	공포	스릴러	코미디	가족	애니메이션	범죄

영화 키워드 :

영화 장르 :

영화 검색

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
var xhr = new XMLHttpRequest();
xhr.withCredentials = true;
xhr.addEventListener("readystatechange", function () {
    if (this.readyState === 4) {
        console.log(this.responseText);
        // 로그 데이터 출력
        // document.getElementById("raw_data").innerHTML = this.responseText;
        var tbl = document.getElementById('tbl0');
        tbl.innerHTML = "<tr><th>영화 제목</th><th>영화 링크</th><th>이미지</th><th>영화 감독</th><th>평점</th></tr>";

        var parsed = JSON.parse(this.responseText);
        for (i in parsed.items) {
            var cur = parsed.items[i];
            tbl.innerHTML += "<tr><td>" + cur.title + "</td><td>" + cur.link + "</td><td>"
                + cur.image + "</td><td>" + cur.director + "</td><td>" + cur.userRating + "</td></tr>";
        }
    }
});

function displayResponse(){
    if(document.getElementById('query').value.length > 30)
    {
        alert("query length error");
        return;
    }

    // 로컬 호스트에 요청을 보냄
    var url = "http://localhost:3000/movie";
    var query = document.getElementById('query').value;
    var genre = document.getElementById('genre').value;
    url = url + "?query="+query + "&genre="+genre
    xhr.open("GET", url);
    xhr.send();
}
```

영화 링크	이미지	영화 감독	평점
https://movie.naver.com/movie/bi/mi/basic.nhn?code=18781	https://ssl.pstatic.net/img/movie/mdi/mit110/0187/18781_P40_110031.jpg	미야자키 하야오	9.26

- AJAX를 이용해서 사용자의 요청에 알맞은 정보를 가지고 올 수 있도록 함
- JSON을 이용해서 OPEN API를 통해 얻은 데이터를 파싱하여 사용함

JavaScript를 활용해 부적절한 입력
(query 길이가 30 초과)에 대한 에러 처리

jQuery/AJAX/JSON 기능 활용

```
localStorage.setItem(memo["date"],JSON.stringify(memo));  
// 결과 출력  
document.getElementById("result").innerHTML += "<h1> 제목: " + memo["title"] + "</h1>" + "날짜: " + memo["date"] + "<br>" + "기분: " + memo["emotion"] + "<br><br>" + memo["content"] ;  
document.getElementById("result").style.border = "1px solid #999";  
}  
// Reset an object in localStorage:  
function reset(){  
    localStorage.clear();  
    document.getElementById("result").innerHTML = "" ;  
    document.getElementById("result").style.border = "none";  
}  
function search(){  
    var key_d = document.getElementById("date2").value;  
    var loc = JSON.parse(localStorage[key_d]);  
    document.getElementById("result").innerHTML += "<h1> 제목: " + loc["title"] + "</h1>" + "날짜: " + loc["date"] + "<br>" + "기분: " + loc["emotion"] + "<br><br>" + loc["content"] ;  
}  
function bin(){  
    document.getElementById("result").innerHTML = "" ;  
    document.getElementById("result").style.border = "none";  
}  
}  
</script>
```

Form을 통해 얻은 값들을
JSON 을 이용하여 다이어리의 제목, 날씨, 기분, 내용을
저장 및 파싱하여 사용

Node.js

Movie List

Find your movie

Movie Genre							
1	2	4	7	11	12	15	16
드라마	판타지	공포	스릴러	코미디	가족	애니메이션	범죄

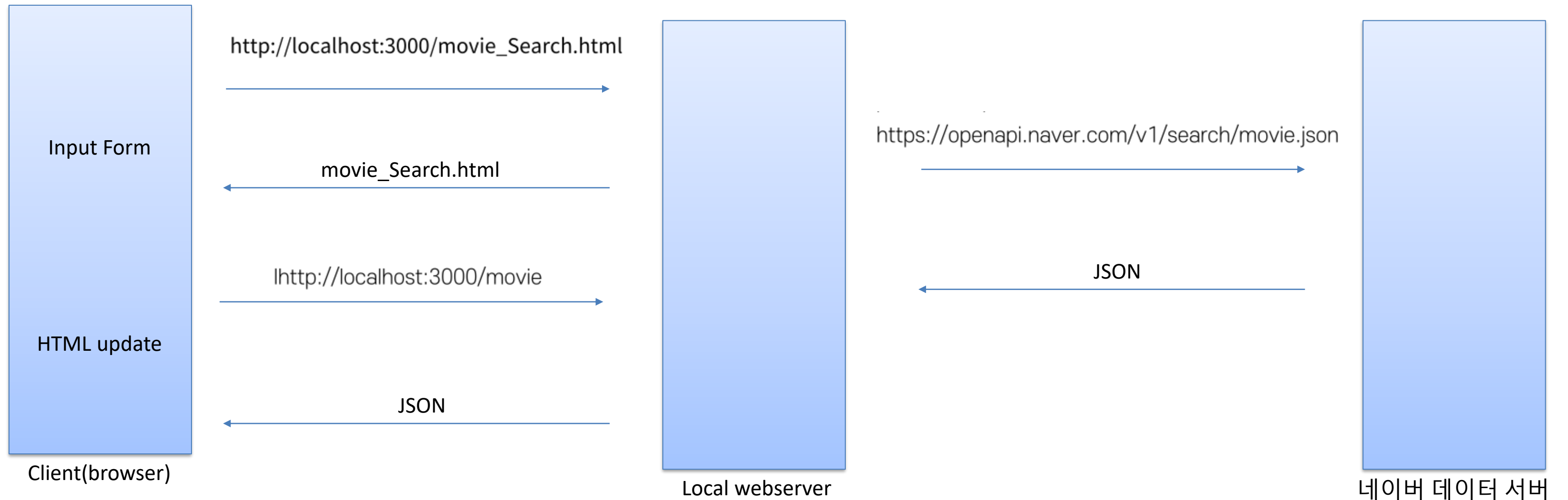
영화 키워드 :

영화 장르 :

영화 검색

영화 제목	영화 링크	이미지	영화 감독	평점
이웃집 토토로	https://movie.naver.com/movie/bi/mi/basic.nhn?code=18781	https://ssl.pstatic.net/img/movie/mdi/mit110/0187/18781_P40_110031.jpg	미야자키 하야오	9.26

Node.js를 활용한 local webserver를 이용하여 API server로 부터 받은 data를 forwarding



추가 구현 기능

- Form의 action을 이용하여 google, youtube 검색 기능
- Form의 데이터를 localStorage에 저장하여 지속적으로 볼 수 있도록 구현
→ 날짜별로 검색이 가능하며, 초기화 기능도 포함



```
<form method=get action="http://www.youtube.com/results" target="_blank" >
  YouTube
  <input type=text name=q size=20% maxlength=255 value="" /> <!-- 유튜브 검색 입력 창 -->
  <input type=submit name=btnG2 value="Search" /> <!-- 검색 버튼 -->
  <p id="weather"></p>
</form>
</nav>
<section id="google">
  <form method=get action="http://www.google.co.kr/search" target="_blank" >
    <input type=text name=q size=100% maxlength=255 value="" /> <!-- 구글 검색 입력 창 -->
    <input type=submit name=btnG value="Search" /> <!-- 검색 버튼 -->
  </form>
```

```
// create an object
function show(){
  for (var i=0; i<5; i++) {
    if (document.getElementsByName("category")[i].checked == true) {
      e = document.getElementsByName("category")[i].value;
    }
  }
  //JSON data
  var memo = {
    title: document.getElementById('fname').value,
    date: document.getElementById('date').value,
    emotion: e,
    content:document.getElementById('memo').value
  };

  localStorage.setItem(memo["date"],JSON.stringify(memo));
  // 결과 출력
  document.getElementById("result").innerHTML += "<h1> 제목: " + memo["title"] + "</h1>" + "날짜: " + memo["date"] + "<br>" + "기분: " + memo["emotion"] + "<br><br>" + memo["content"] ;
  document.getElementById("result").style.border = "1px solid #999";
}

// Reset an object in localStorage:
function reset(){
  localStorage.clear();
  document.getElementById("result").innerHTML = "" ;
  document.getElementById("result").style.border = "none";
}

function search(){
  var key_d = document.getElementById("date2").value;
  var loc = JSON.parse(localStorage[key_d]);
  document.getElementById("result").innerHTML += "<h1> 제목: " + loc["title"] + "</h1>" + "날짜: " + loc["date"] + "<br>" + "기분: " + loc["emotion"] + "<br><br>" + loc["content"] ;
}

function bin(){
  document.getElementById("result").innerHTML = "" ;
  document.getElementById("result").style.border = "none";
}
```


02.

웹 사이트 사용법

1. 터미널에서 node index.js 입력

2. 웹 브라우저에서 http://localhost:3000/main_page.html 접속

- **Youtube 검색** : 원하는 검색어를 입력하고, Search 버튼을 입력하면 관련 동영상 검색이 바로 가능합니다.



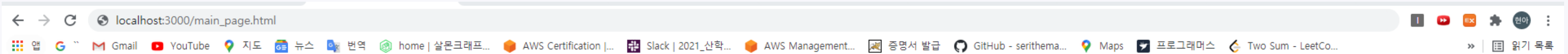
- **다른 HTML 페이지 이동**

왼쪽부터 차례대로
movie_Search.html /
todo_list.html/add_schedule.html/Diary.html/StopWatch.html
각 페이지 링크를 입력하면 해당 페이지로 이동합니다.

- **실시간 날씨 출력**: 현재 부산의 기온과 습도를 OPEN API를 통해서 출력합니다.

- **Google 검색** : 원하는 검색어를 입력하고, Search 버튼을 입력하면 구글링이 바로 가능합니다.

01. Movie_Search.html



Movie List

Find your movie

Movie Genre

1	2	4	7	11	12	15	16
드라마	판타지	공포	스릴러	코미디	가족	애니메이션	범죄

영화 키워드 : 토토로

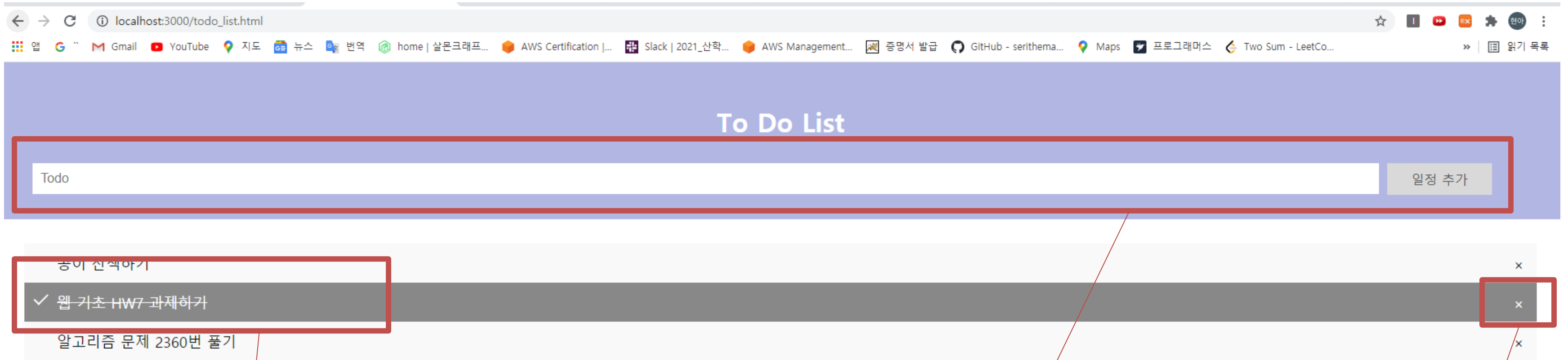
영화 장르 : 15

영화 검색

영화 제목	영화 링크	이미지	영화 감독	평점
이웃집 토토로	https://movie.naver.com/movie/bi/mi/basic.nhn?code=18781	https://ssl.pstatic.net/imgmovie/mdi/mit110/0187/18781_P40_110031.jpg	미야자키 하야오	9.26

- **영화 검색** : 원하는 영화 관련된 키워드나 장르(위의 Movie Genre table 참고)를 입력하여 영화 검색 버튼을 누르면 아래에 검색 결과를 출력합니다.

02. todo_list.html



- **할 일 체크:** 업무 수행을 완료한 항목은 클릭을 하면 Check 표시가 됩니다.
- **오늘의 할 일 추가 :** 오늘 해야 할 일정 등을 입력한 뒤
일정 추가 버튼을 누르면 아래의 리스트에 추가됩니다.
- **할 일 삭제:** 더 이상 필요없는 항목은 x 표시를 클릭하면
리스트에서 삭제된다.

03. Diary.html

Hyun's Diary

Do Your Best!

Add Diary :)

Title :

Date :

Emotion
행복 ☒ 슬픔 ☐ 기쁨 ☐ 화남 ☐ 놀람 ☐

Content

비우기

취소

저장

4. 일기 검색: 작성한 날짜를 선택하면 해당 날짜에 대한 일기가 화면에 출력됩니다.

Search Diary :(

▼ 원하는 일기 날짜를 선택하세요 ▼
Date :

초기화

홈화면

검색

제목: 오늘은 나의 생일

날짜: 2021-06-02
기분: happy

오늘은 드디어 나의 생일이다!! 맛있는 것도 마니 먹고 무지 행복한 날이었다!!

1. 일기 작성: 해당 날짜에 일기 제목, 감정, 일기 본문을 입력한 후, 저장 버튼을 누르면 일기가 저장됩니다. 취소 버튼을 누르면 입력 내용이 리셋됩니다.

2. 출력 화면 비우기 : 저장 버튼을 누를 때 출력된 아래의 데이터 화면을 비우기 위해서 비우기 버튼을 클릭합니다.

3. 초기화: 초기화 버튼을 누르면 로컬 스토리지와 아래의 데이터 화면이 초기화 됩니다.

5. 홈 버튼 : 홈화면 버튼을 누르면 메인 화면으로 돌아갑니다.

04. Stopwatch.html

localhost:3000/StopWatch/stopwatch.html

- **Study Start** : 버튼을 누르면 스톱워치가 시작됩니다.
- **Study Stop** : 버튼을 누르면 스톱워치가 일시 중지 됩니다.
- **Time Reset** : 버튼을 누르면 스톱워치가 리셋됩니다.

6월 6일 13:28:34

00:00:08

Study Start Study Stop Time Reset

00:00:08

- 버튼을 누르면 중지될 때 기록된 데이터가 삭제됩니다.

X

THANK

YOU