

# 인터넷과 웹 기초 팀 프로젝트 최종보고서

학번: 201914116

학과: 정보컴퓨터공학부

이름: 김휘수

# 목차

## I. 최소 개발 기준

- A. 서비스 기능 3가지 이상 구현
- B. OPEN API 2개 이상 활용
- C. 서로 다른 <input> 3개 이상 활용
- D. List 및 Table 활용
- E. pseudo-class 2개 이상, attribute selector 2개 이상, combinator 5개 이상 활용
- F. semantic tag 활용한 웹 layout 구성
- G. DOM object 5개 이상 활용
- H. input 태그 요소 내 입력값 formatting 구현
- I. 소스 코드에 각 함수 및 최소 개발 기준 별로 기능 설명하는 주석

## II. 최소 개발 기준 이외에 추가 개발 항목

- A. 웹 페이지 내부 네비게이션
- B. 네비게이션에 애니메이션 추가

## III. 웹 사이트 동작 시나리오

- A. 오늘의 날씨, 주가 정보 받아오기
- B. 할 일 리스트 추가하기
- C. ChatGPT 기반 챗봇과 대화하기

## IV. 보완해야 할 부분

- A. 화면 크기에 따른 깨짐 현상
- B. 화면의 구성과 id와 class의 통일성
- C. 다양한 API 요청 방법

**최소 개발 기준**


## A. 서비스 기능 3가지 이상 구현

My Homepage


부산대학교    학생지원시스템    정보컴퓨터공학부    PLATO    공학교육혁신센터

Today


check today's various information!



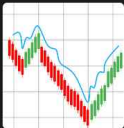
**코스피**  
고가: 2612.39 저가: 2595.33  
시가: 2609.21 종가: 2610.85  
등락률: -0.18%  
2023. 06. 08. 기준



**코스닥**  
고가: 879.44 저가: 871.5  
시가: 878.86 종가: 876.13  
등락률: -0.52%  
2023. 06. 08. 기준



**삼성전자**  
고가: 70900 저가: 70000  
시가: 70400 종가: 70900  
등락률: -0.14%  
2023. 06. 08. 기준



**펜오션**  
고가: 5380 저가: 5220  
시가: 5230 종가: 5310  
등락률: 1.72%  
2023. 06. 08. 기준

Yangsan  
24.71°C  
Clouds

Sun, 2023. 06. 11.

14 : 41 : 09

good Afternoon, 휘수

Today

file:///C:/Users/user/Desktop/과제/인터넷과웹기초/Term%20Project/index.html?#today

24°C  
비 약간

검색

오후 2:41  
2023-06-11



```

async function getStockInfo(url, i) {
  await fetch(url).then(response => response.json()).then(data => {
    let stock_info = new Array(5);
    const baseDate = data.response.body.items.item[0].basDt
    const stockName = data.response.body.items.item[0].itmsNm
    const highPrice = data.response.body.items.item[0].hipr
    const lowPrice = data.response.body.items.item[0].lopr
    const marketPrice = data.response.body.items.item[0].mkp
    const closePrice = data.response.body.items.item[0].clpr
    const rate = data.response.body.items.item[0].fltRt

    stock_info[0] = `${baseDate.substr(0,4)}. ${baseDate.substr(4,2)}. ${baseDate.substr(6,2)}. 기준`
    stock_info[1] = `${stockName}`
    stock_info[2] = `고가: ${highPrice} 저가: ${lowPrice}`
    stock_info[3] = `시가: ${marketPrice} 종가: ${closePrice}`
    stock_info[4] = `등락률: ${setRateColor(rate, i)}%`

    document.getElementsByClassName("base_date")[i].innerHTML = stock_info[0]
    document.getElementsByClassName("stock_name")[i].innerHTML = stock_info[1]
    document.getElementsByClassName("high_low")[i].innerHTML = stock_info[2]
    document.getElementsByClassName("market_close")[i].innerHTML = stock_info[3]
    document.getElementsByClassName("rate")[i].innerHTML = stock_info[4]
  }).catch(function(error) {
    console.error(error);
  });
}

```

1. 주식, 금융지수 API 활용한 오늘의 시세 정보 출력
2. 사용자로부터 주식 종목을 입력받아 해당하는 주식의 정보 출력, 등락률에 따른 텍스트 색상 변경 구현



```
function setClock() {  
    const now = new Date();  
    time.innerHTML = `${now.getHours().toString().padStart(2, "0")} :  
        ${now.getMinutes().toString().padStart(2, "0")} :  
        ${now.getSeconds().toString().padStart(2, "0")}`;  
}  
setInterval(setClock, 1000)  
  
function hourToTime(hour) {  
    if(hour >= 18) {  
        return "Evening";  
    } else if(hour >=12) {  
        return "Afternoon";  
    } else if(hour >= 6) {  
        return "Moring";  
    } else return "Night";  
}  
  
function dateToday(day) {  
    switch(day) {  
        case 0:  
            return "Sun";  
            break;  
        ...  
    }  
}
```

3. setInterval 과 Date class를 이용한 날짜와 시각 출력, 시간에 따른 인사말 출력

Yangsan  
24.71°C  
Clouds

```
async function onGeoOk(position) {
  const lat = position.coords.latitude;
  const lon = position.coords.longitude;
  const url = `https://api.openweathermap.org/data/2.5/weather
              ?lat=${lat}&lon=${lon}&appid=${API_KEY_WEATHER}&units=metric`;
  await fetch(url).then(response => response.json()).then(data => {
    const city = document.getElementById("city");
    const weather = document.getElementById("weather");
    const temp = document.getElementById("temp");
    city.innerHTML = data.name;
    weather.innerHTML = data.weather[0].main;
    temp.innerHTML = `${data.main.temp}°C`;
    if(data.weather[0].main == "Rain") {
      let template = `<span class="weather_info" id="city">Don't forget to
                      bring your umbrella!</span>`;
      weatherInfoContainer.insertAdjacentHTML("beforeend", template)
    }
  }).catch(function(error) {
    console.error(error);
  });
}

function onGeoError() {
  alert("error");
}

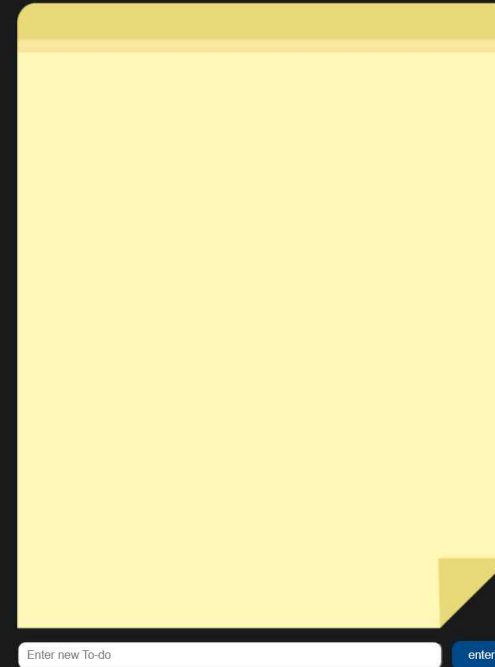
navigator.geolocation.getCurrentPosition(onGeoOk, onGeoError);
```

4. navigator.geolocation.getCurrentPosition()과 날씨 API를 통해 위치, 기온, 날씨 출력
5. 날씨가 "Rain"이면 Don't forget to bring your umbrella! 문구를 하단에 출력

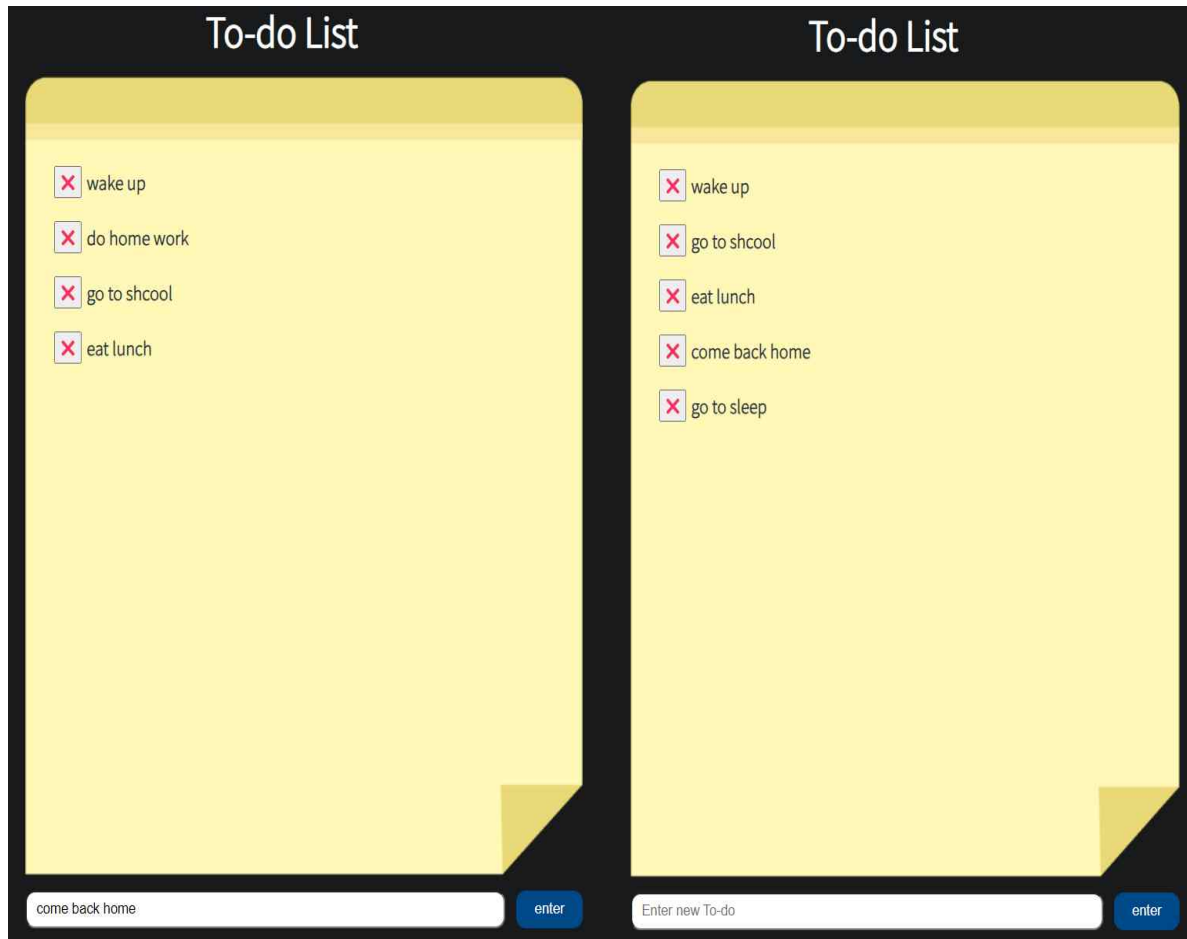
check today's schedule!

## To-do List

	Mon	Tue	Wed	Thu	Fri
09:00~10:15	Introduction to Internet and Web		Introduction to Internet and Web		
10:30~11:45	Logic Circuit Design		Logic Circuit Design		
13:30~14:45		Engineering Linear Algebra		Engineering Linear Algebra	
15:00~16:15	Computer Architecture	Introduction to UNIX Programming	Computer Architecture	Introduction to UNIX Programming	
16:30~17:45		Operating Systems		Operating Systems	
18:00~22:00	Computer Software Design&Lab				





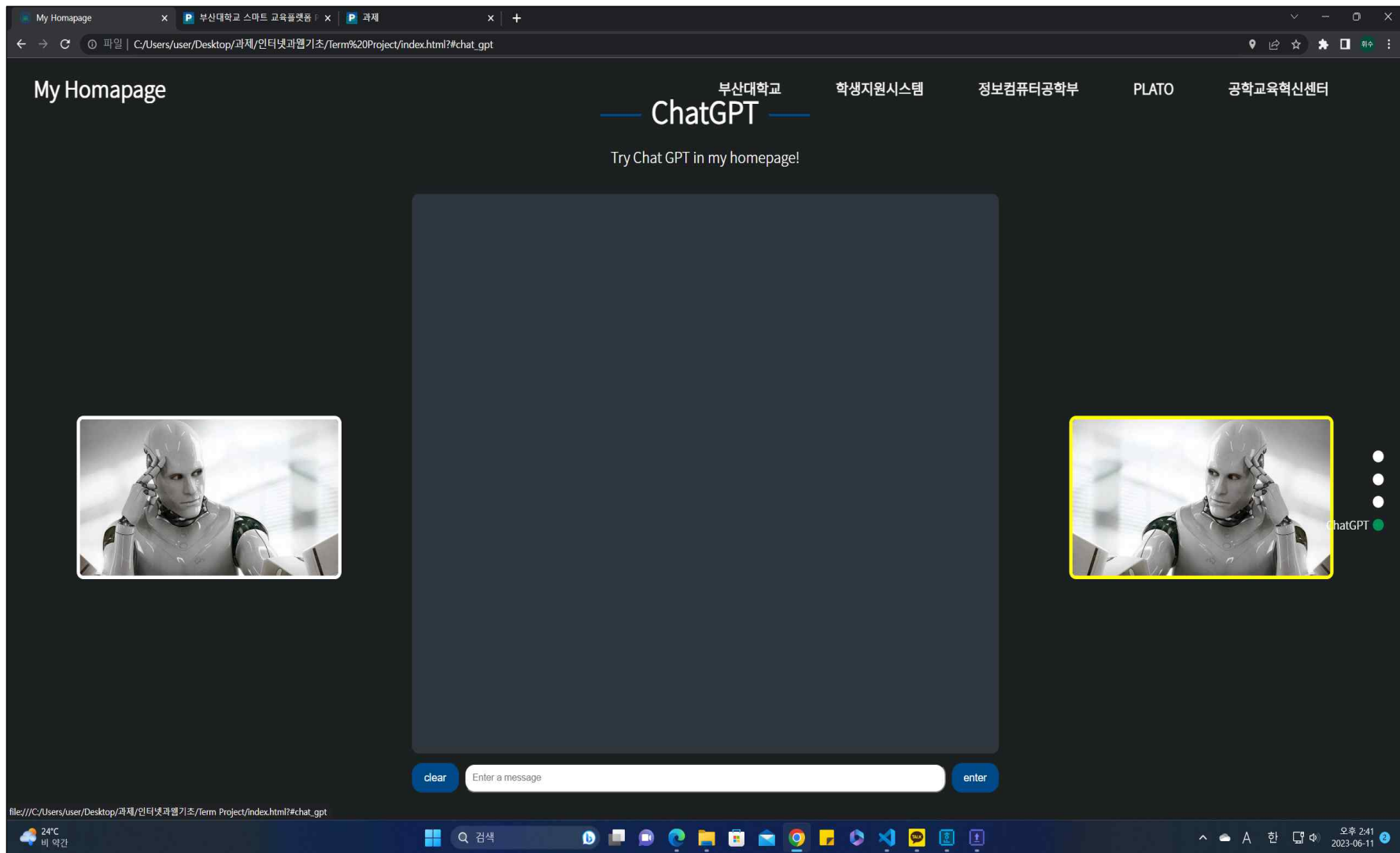


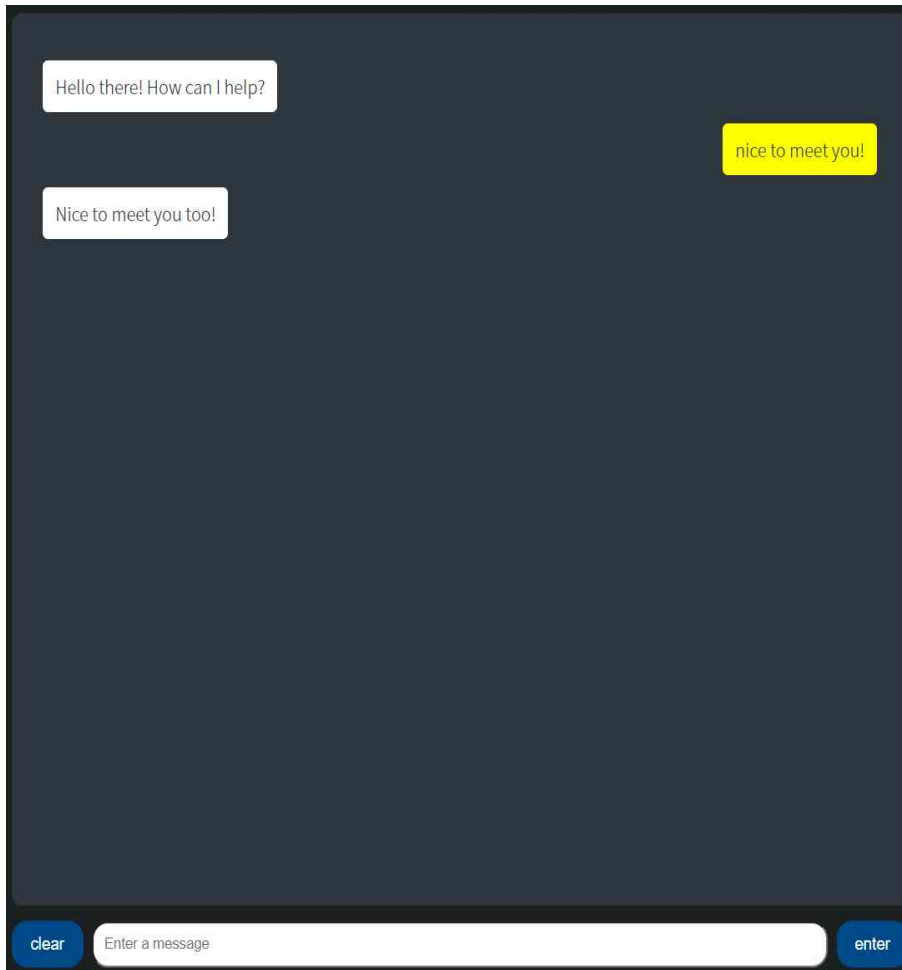
```
function addToDo(newTodo) {
    const li = document.createElement("li")
    const span = document.createElement("span")
    const button = document.createElement("button")
    li.setAttribute("class", "todo_li")
    span.setAttribute("class", "todo_span")
    button.setAttribute("class", "todo_button")
    span.innerText = newTodo
    button.innerText = " "
    button.addEventListener("click", removeToDo)
    li.appendChild(button)
    li.appendChild(span)
    toDoList.appendChild(li)
}

function removeToDo(event) {
    const li = event.target.parentElement
    li.remove()
}

function toDoSubmit() {
    const newTodo = toDoInput.value
    if(newTodo !== "") {
        toDoInput.value = ""
        addToDo(newTodo)
    }
}
```

6. To-do List에 할 일 추가 및 삭제 기능 구현





```

async function getMessage() {
  const myMessage = document.getElementById("chat_input").value;
  let template = `

7. ChatGPT API를 활용한 챗봇 구현



8. 보낸 메시지와 받은 메시지를 시각적으로 구현, 메시지 삭제 기능 구현


```

## B. OPEN API 2개 이상 활용



```
async function getStockInfo(url, i) {
  await fetch(url).then(response => response.json()).then(data => {
    let stock_info = new Array(5);
    const baseDate = data.response.body.items.item[0].basDt
    const stockName = data.response.body.items.item[0].itmsNm
    const highPrice = data.response.body.items.item[0].hipr
    const lowPrice = data.response.body.items.item[0].lopr
    const marketPrice = data.response.body.items.item[0].mkp
    const closePrice = data.response.body.items.item[0].clpr
    const rate = data.response.body.items.item[0].fltRt

    stock_info[0] = `${baseDate.substr(0,4)}. ${baseDate.substr(4,2)}. ${baseDate.substr(6,2)}. 기준`
    stock_info[1] = `${stockName}`
    stock_info[2] = `고가: ${highPrice} 저가: ${lowPrice}`
    stock_info[3] = `시가: ${marketPrice} 종가: ${closePrice}`
    stock_info[4] = `등락률: ${setRateColor(rate, i)}%`

    document.getElementsByClassName("base_date")[i].innerHTML = stock_info[0]
    document.getElementsByClassName("stock_name")[i].innerHTML = stock_info[1]
    document.getElementsByClassName("high_low")[i].innerHTML = stock_info[2]
    document.getElementsByClassName("market_close")[i].innerHTML = stock_info[3]
    document.getElementsByClassName("rate")[i].innerHTML = stock_info[4]
  }).catch(function(error) {
    console.error(error);
  });
}
```

1. 주식 API를 활용하여 주식 정보 출력, 사용자 입력 주식 정보 출력
2. 금융 지수 API를 활용하여 KOSPI, KOSDAQ 정보 출력

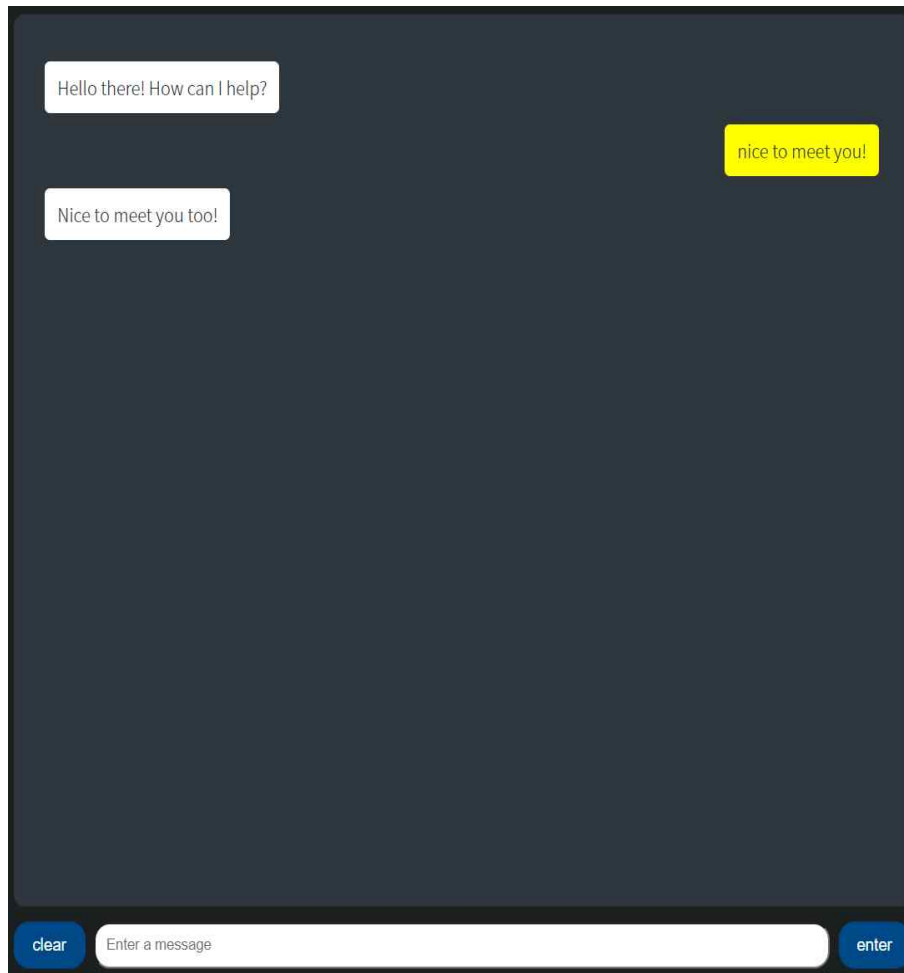
Yangsan  
24.71°C  
Clouds

```
async function onGeoOk(position) {
  const lat = position.coords.latitude;
  const lon = position.coords.longitude;
  const url = `https://api.openweathermap.org/...
    ?lat=${lat}&lon=${lon}&appid=${API_KEY_WEATHER}&units=metric`;
  await fetch(url).then(response => response.json()).then(data => {
    const city = document.getElementById("city");
    const weather = document.getElementById("weather");
    const temp = document.getElementById("temp");
    city.innerHTML = data.name;
    weather.innerHTML = data.weather[0].main;
    temp.innerHTML = `${data.main.temp}°C`;
    if(data.weather[0].main == "Rain") {
      let template = `<span class="weather_info" id="city">Don't forget to
        bring your umbrella!</span>`;
      weatherInfoContainer.insertAdjacentHTML("beforeend", template)
    }
  }).catch(function(error) {
    console.error(error);
  });
}

function onGeoError() {
  alert("error");
}

navigator.geolocation.getCurrentPosition(onGeoOk, onGeoError);
```

3. 날씨 API를 활용하여 오늘의 날씨, 기온 출력

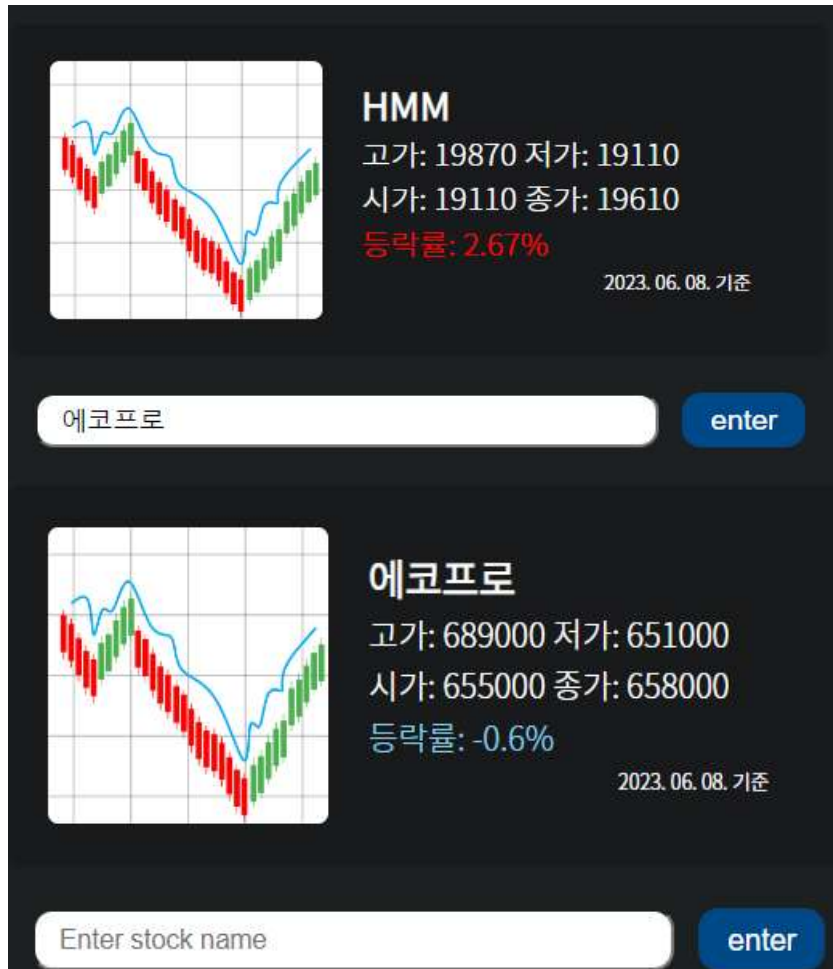


```
async function getMessage() {
  const myMessage = document.getElementById("chat_input").value;
  let template = `

#### 4. ChatGPT API를 활용하여 챗봇 구현


```

### C. 서로 다른 <input> 3개 이상 활용



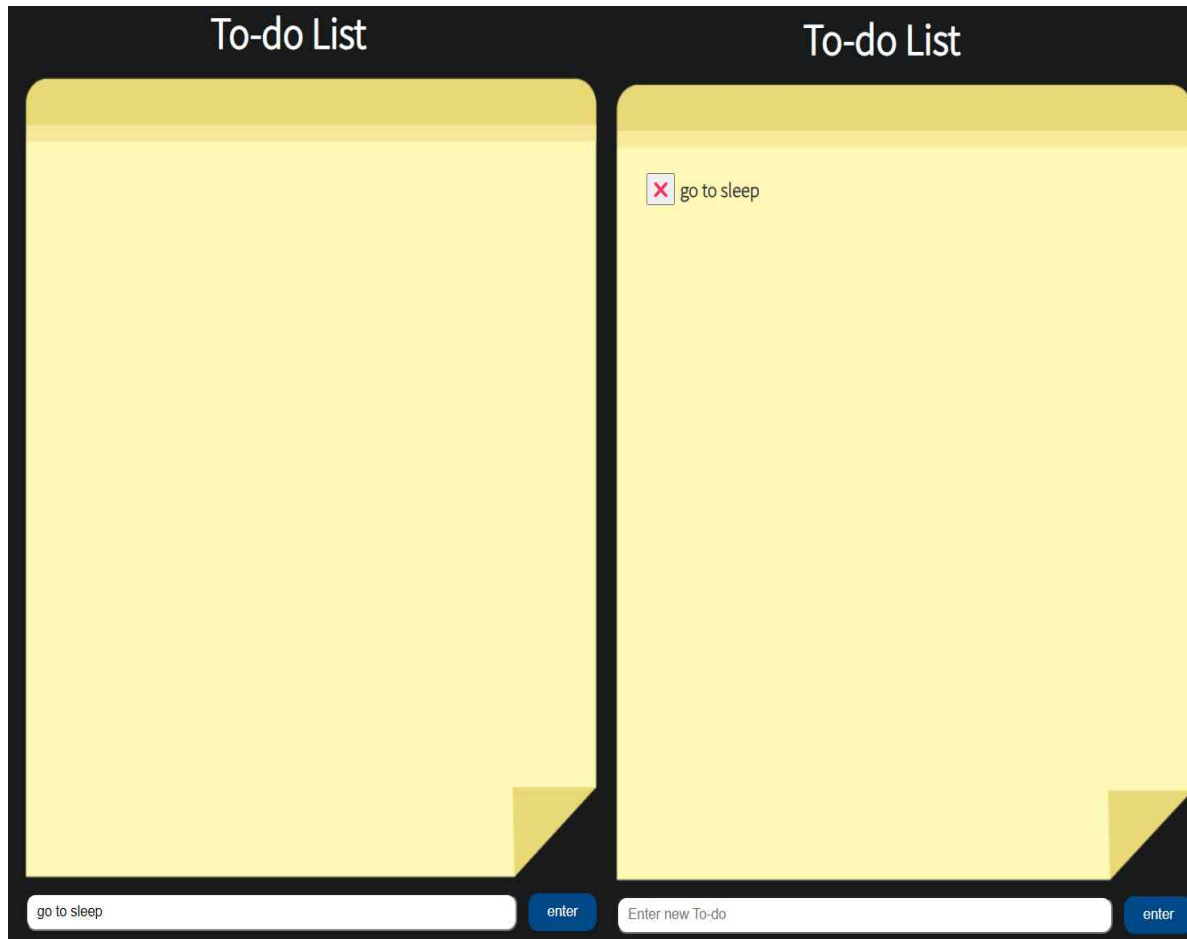
```
async function getStockInfo(url, i) {
  await fetch(url).then(response => response.json()).then(data => {
    let stock_info = new Array(5);
    const baseDate = data.response.body.items.item[0].basDt
    const stockName = data.response.body.items.item[0].itmsNm
    ...

    stock_info[0] = `${baseDate.substr(0,4)}.
                      ${baseDate.substr(4,2)}. ${baseDate.substr(6,2)}. 기준`
    ...

    document.getElementsByClassName("base_date")[i].innerHTML = stock_info[0]
    ...
  }).catch(function(error) {
    console.error(error);
  });
}

stockBtn.addEventListener("click", function() {
  const myStockName = myStock.value
  const MyUrl = `https://apis.data.go.kr/ ... itmsNm=${myStockName}`
  myStock.value = ""
  getStockInfo(MyUrl, 3)
})
```

1. 주식종목을 입력받는 input



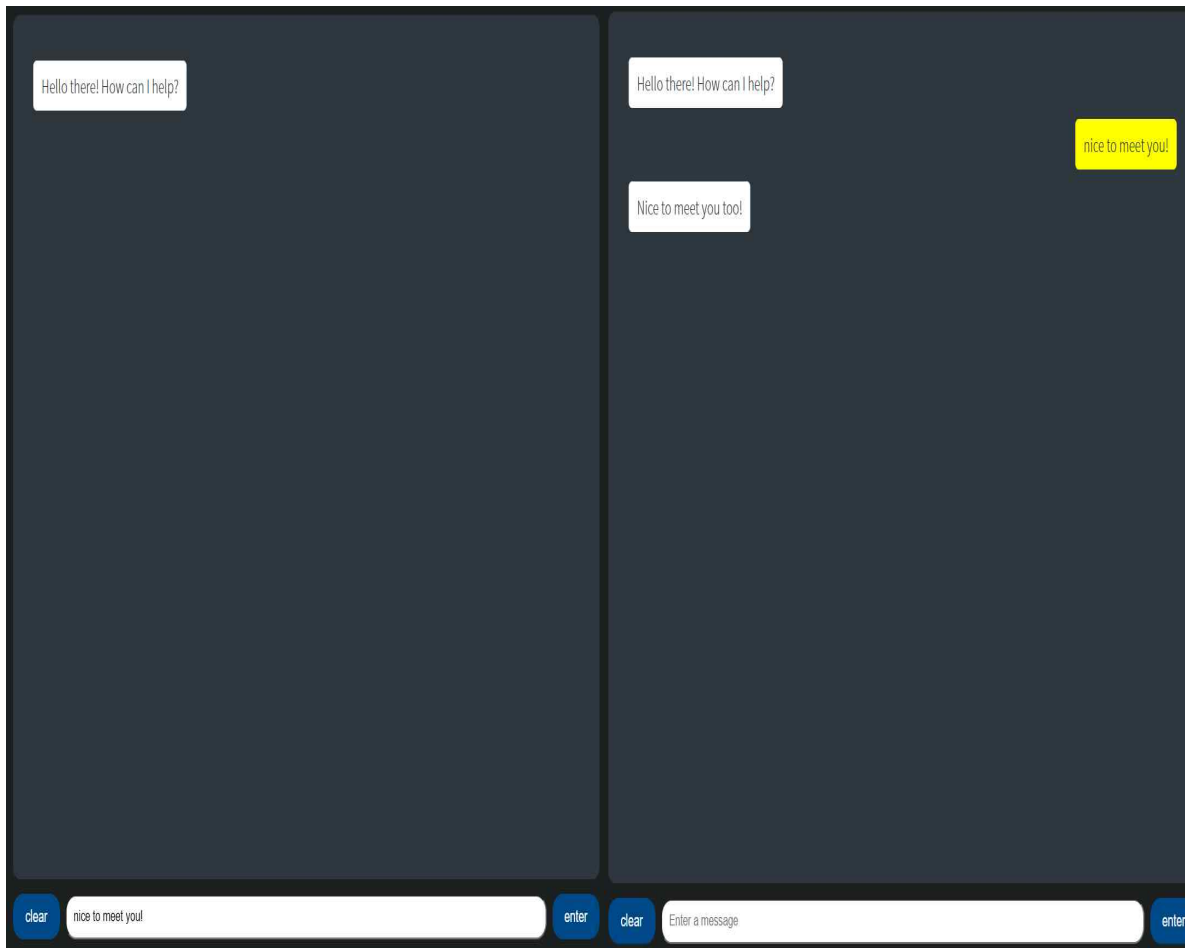
```
function addToDo(newTodo) {
  const li = document.createElement("li")
  const span = document.createElement("span")
  const button = document.createElement("button")
  li.setAttribute("class", "todo_li")
  span.setAttribute("class", "todo_span")
  button.setAttribute("class", "todo_button")
  span.innerText = newTodo
  button.innerText = " "
  button.addEventListener("click", removeToDo)
  li.appendChild(button)
  li.appendChild(span)
  toDoList.appendChild(li)
}

function removeToDo(event) {
  const li = event.target.parentElement
  li.remove()
}

function toDoSubmit() {
  const newTodo = toDoInput.value
  if(newTodo !== "") {
    toDoInput.value = ""
    addToDo(newTodo)
  }
}
```

2. 추가할 할 일을 입력받는 input





```

async function getMessage() {
  const myMessage = document.getElementById("chat_input").value;
  let template = `<div class="line">
    <span class="chat_box mine">${myMessage}</span></div>`;
  insetrtMessage(template)

  const options = {
    method: "POST",
    headers: {
      "Authorization": `Bearer ${API_KEY_GPT}`,
      "Content-Type": "application/json"
    },
    body: JSON.stringify({
      model: "text-davinci-003",
      prompt: myMessage,
      max_tokens: 100
    })
  }
  document.getElementById("chat_input").value = ""
  try {
    const response =
      await fetch("https://api.openai.com/v1/completions", options)
    const data = await response.json()
    let template = `<div class="line">
      <span class="chat_box">${data.choices[0].text}</span></div>`
    insetrtMessage(template)
  } catch(error) {
    console.error(error)
  }
}

function insetrtMessage(template) {
  document.getElementById("chat").insertAdjacentHTML("beforeend",
    template)
}

```

### 3. chatGPT에게 보낼 메시지를 받는 input

## D. List 및 Table 활용

3 - 1					
	Mon	Tue	Wed	Thu	Fri
09:00~10:15	Introduction to Internet and Web		Introduction to Internet and Web		
10:30~11:45	Logic Circuit Design		Logic Circuit Design		
13:30~14:45		Engineering Linear Algebra		Engineering Linear Algebra	
15:00~16:15	Computer Architecture	Introduction to UNIX Programming	Computer Architecture	Introduction to UNIX Programming	
16:30~17:45		Operating Systems		Operating Systems	
18:00~22:00	Computer Software Design&Lab				

...

```
<div id="time_table_container">
```

```
  <div id="time_table">
```

```
    <!--<table> 활용-->
```

```
      <table>...
```

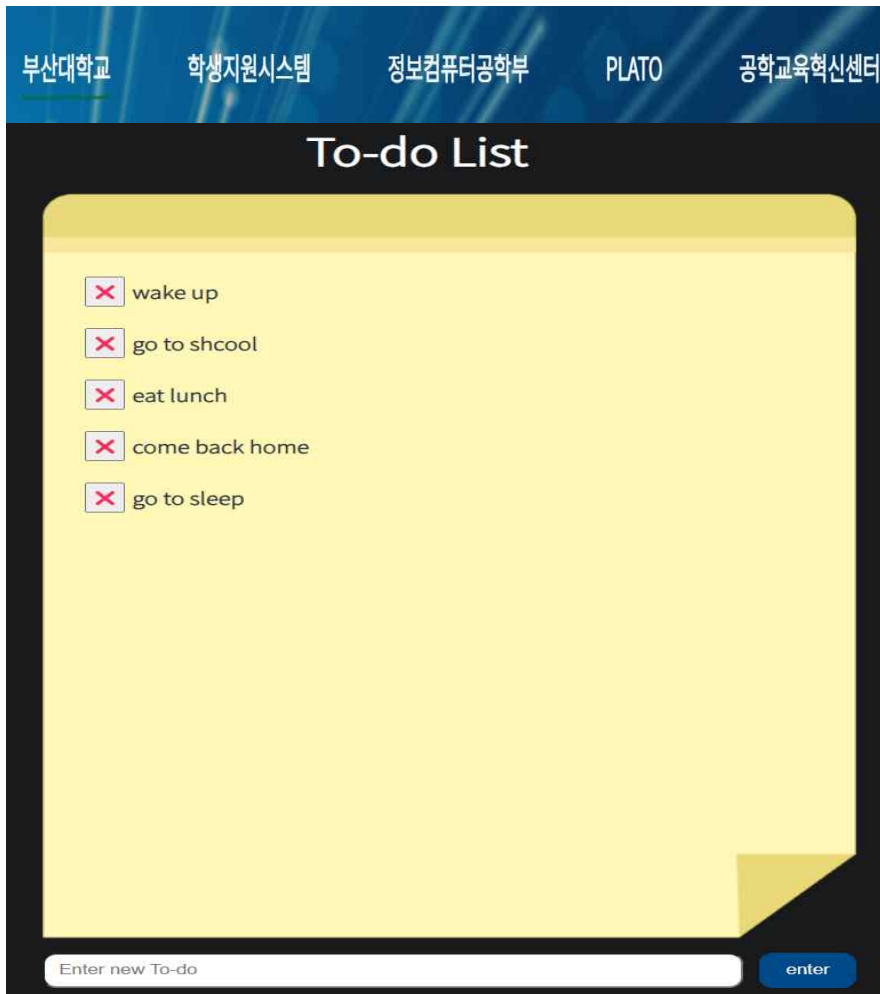
```
    </table>
```

```
  </div>
```

```
</div>
```

...

1. <table>태그를 활용한 시간표 구현



```
<div id="navigation_main">
  <div title="Welcome!" id="logo">My Homapage</div>
  <!--<list> 활용-->
  <ul>...
  </ul>
</div>

<div id="todo_list">
  <!--<list> 활용-->
  <ul id="list">...
  </ul>
</div>
```

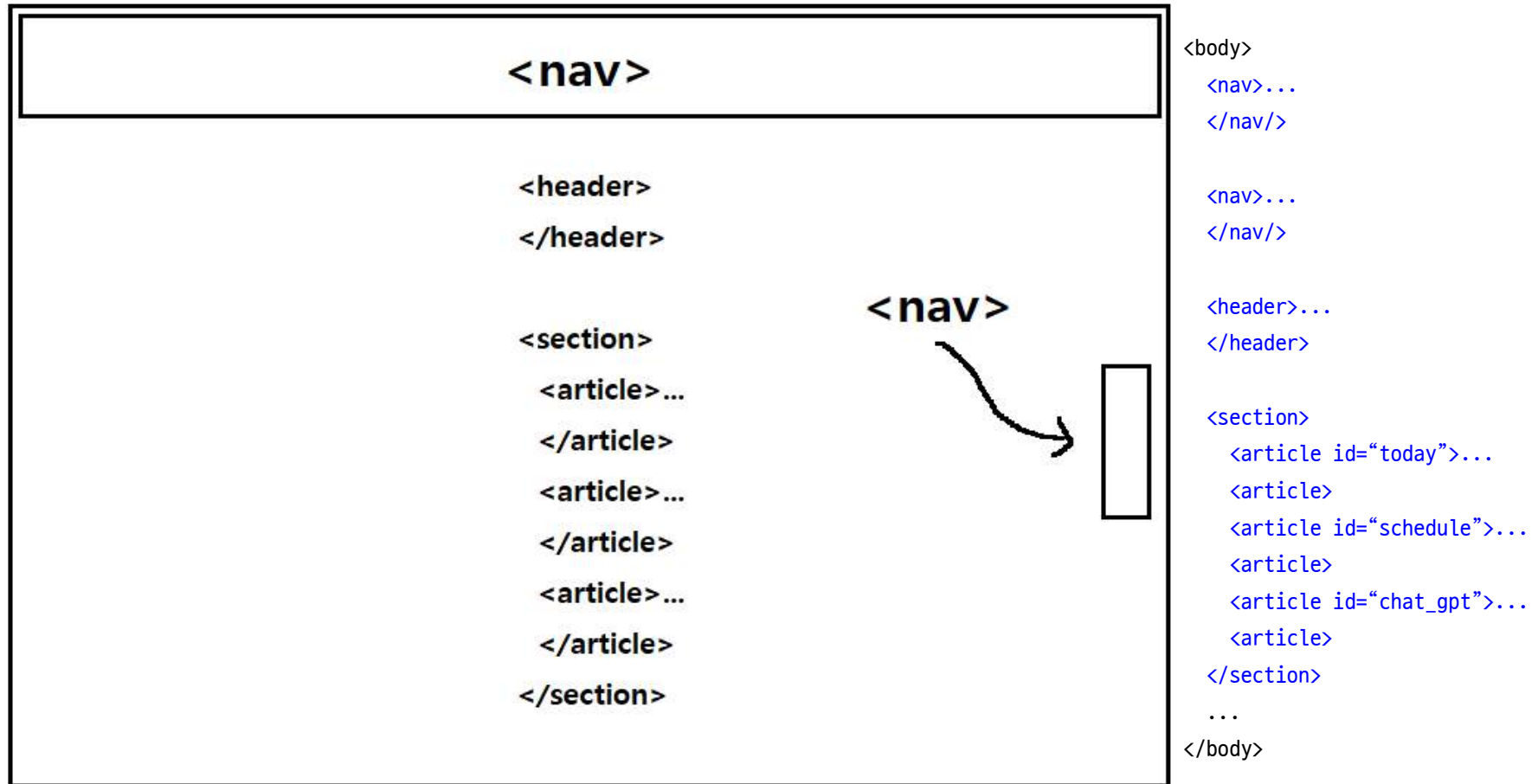
2. <ul>태그를 활용한 네비게이션 내 링크 리스트와 To-do list 구현

## E. pseudo-class 2개 이상, attribute selector 2개 이상, combinator 5개 이상 활용

<pre> /* using pseudo-class 1/2 */ #navigation_main:hover {     background-color: rgba( 0, 0, 0, 0.5 ); }  /* using pseudo-class 2/2 */ #navigation_main a:hover {     border-bottom: 3px solid #026e3c; } </pre>	<pre> /* using attribute selector 1/2 */ a[target = _blank] {     border-bottom: 3px solid rgba(0, 0, 0, 0);     transition: border-bottom 0.25s;     padding-bottom: 5px; }  /* using attribute selector 2/2 */ div[title = "Welcome!"] {     display: flex;     align-items: center;     font-size: 30px;     font-weight: bold; } </pre>	<pre> /* using combinator 1/5 */ #navigation_main ul {     padding: 30px;     display: flex;     justify-content: center; }  /* using combinator 2/5 */ #navigation_main li {     font-weight: 600;     font-size: 20px;     margin: 0px 40px; }  ...  /* using combinator 5/5 */ .navigation_side_container span {     color: white;     visibility: hidden; } </pre>
---	---	--

전부 navigation.css 파일에 구현

## F. semantic tag 활용한 웹 layout 구성



<nav>, <header>, <section>, <article> 등의 semantic tag를 활용하여 html 작성

## G. DOM object 5개 이상 활용



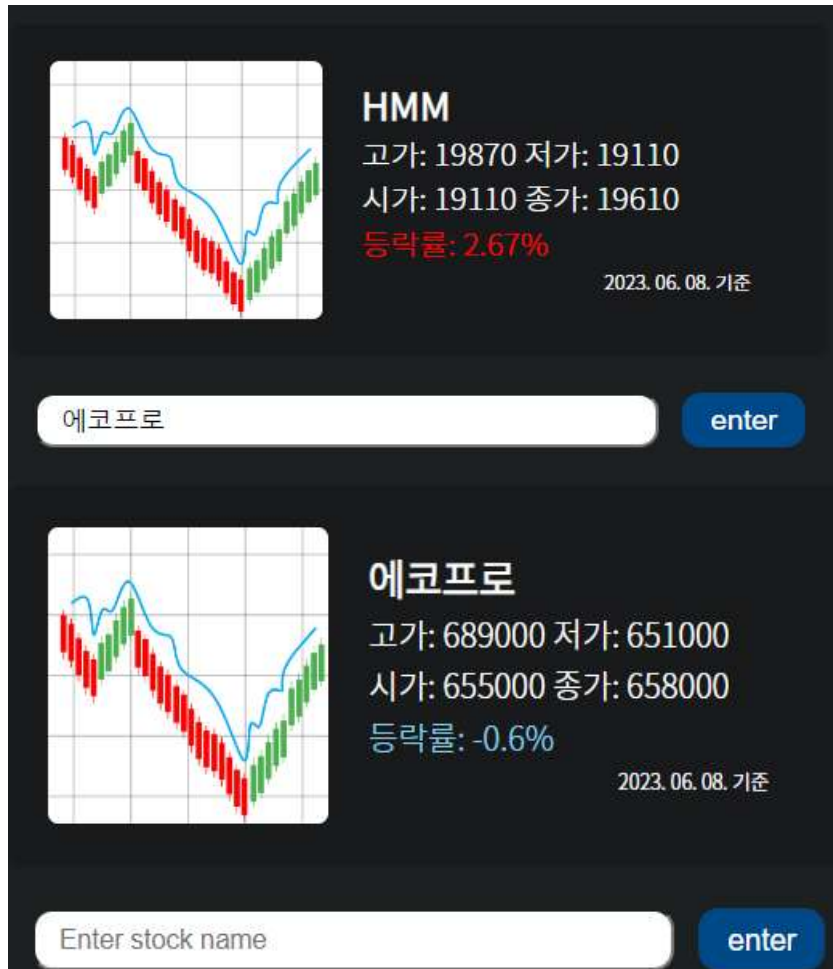
주식 정보를 받아와서 5개의 span의 DOM Object들을 활용하여 출력

```
async function getStockInfo(url, i) {
  await fetch(url).then(response => response.json()).then(data => {
    let stock_info = new Array(5);
    const baseDate = data.response.body.items.item[0].basDt
    const stockName = data.response.body.items.item[0].itmsNm
    const highPrice = data.response.body.items.item[0].hipr
    const lowPrice = data.response.body.items.item[0].lopr
    const marketPrice = data.response.body.items.item[0].mkp
    const closePrice = data.response.body.items.item[0].clpr
    const rate = data.response.body.items.item[0].fltRt

    stock_info[0] = `${baseDate.substr(0,4)}.
      ${baseDate.substr(4,2)}. ${baseDate.substr(6,2)}. 기준`
    stock_info[1] = `${stockName}`
    stock_info[2] = `고가: ${highPrice} 저가: ${lowPrice}`
    stock_info[3] = `시가: ${marketPrice} 종가: ${closePrice}`
    stock_info[4] = `등락률: ${setRateColor(rate, i)}%`

    document.getElementsByClassName("base_date")[i].innerHTML = stock_info[0]
    document.getElementsByClassName("stock_name")[i].innerHTML = stock_info[1]
    document.getElementsByClassName("high_low")[i].innerHTML = stock_info[2]
    document.getElementsByClassName("market_close")[i].innerHTML = stock_info[3]
    document.getElementsByClassName("rate")[i].innerHTML = stock_info[4]
  }).catch(function(error) {
    console.error(error);
  });
}
```

## H. input 태그 요소 내 입력값 formatting 구현



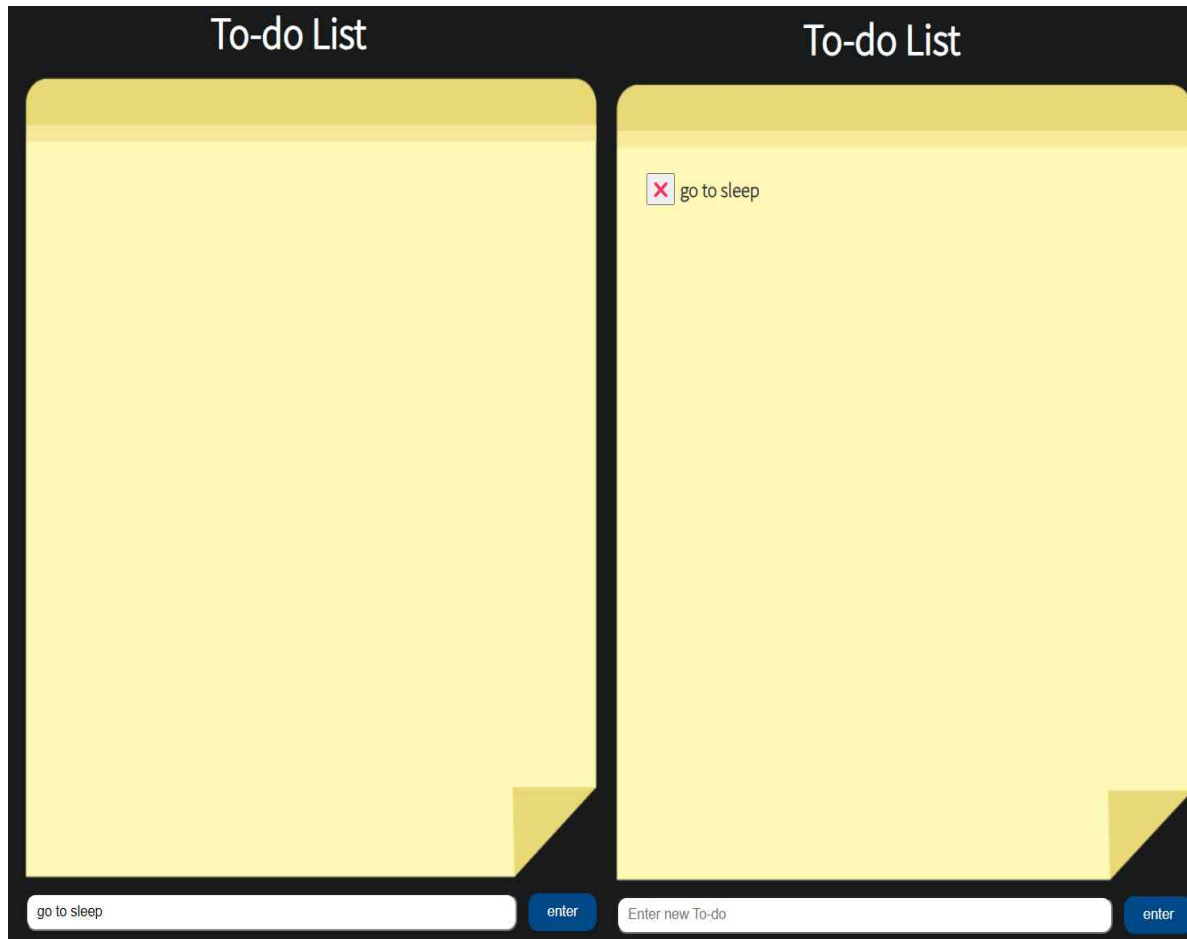
```
async function getStockInfo(url, i) {
  await fetch(url).then(response => response.json()).then(data => {
    let stock_info = new Array(5);
    const baseDate = data.response.body.items.item[0].basDt
    const stockName = data.response.body.items.item[0].itmsNm
    ...

    stock_info[0] = `${baseDate.substr(0,4)}.
                      ${baseDate.substr(4,2)}. ${baseDate.substr(6,2)}. 기준`
    ...

    document.getElementsByClassName("base_date")[i].innerHTML = stock_info[0]
    ...
  }).catch(function(error) {
    console.error(error);
  });
}

stockBtn.addEventListener("click", function() {
  const myStockName = myStock.value
  const MyUrl = `https://apis.data.go.kr/ ... itmsNm=${myStockName}`
  myStock.value = ""
  getStockInfo(MyUrl, 3)
})
```

1. 사용자가 입력한 주식의 정보 출력



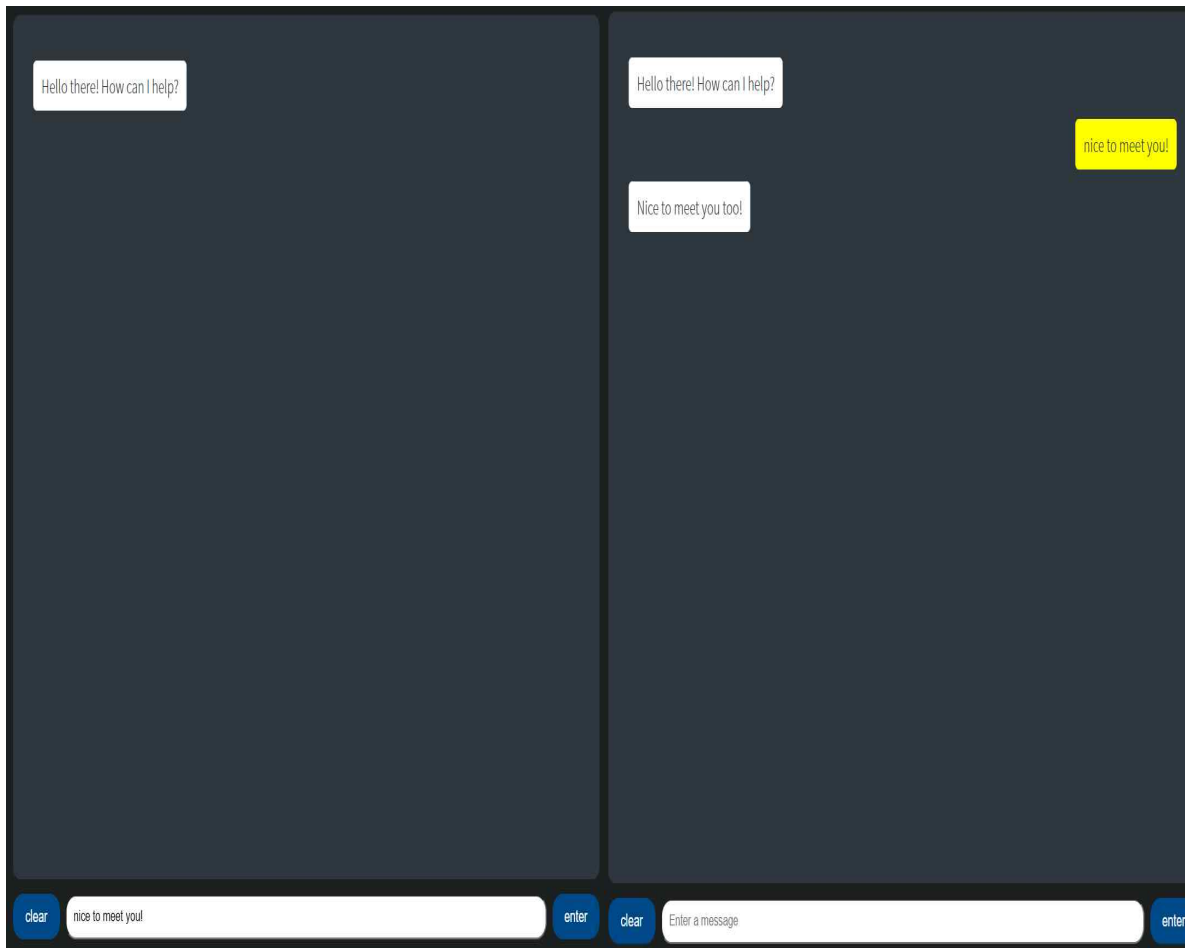
```
function addToDo(newTodo) {
  const li = document.createElement("li")
  const span = document.createElement("span")
  const button = document.createElement("button")
  li.setAttribute("class", "todo_li")
  span.setAttribute("class", "todo_span")
  button.setAttribute("class", "todo_button")
  span.innerText = newTodo
  button.innerText = " "
  button.addEventListener("click", removeToDo)
  li.appendChild(button)
  li.appendChild(span)
  toDoList.appendChild(li)
}

function removeToDo(event) {
  const li = event.target.parentElement
  li.remove()
}

function toDoSubmit() {
  const newTodo = toDoInput.value
  if(newTodo !== "") {
    toDoInput.value = ""
    addToDo(newTodo)
  }
}
```

## 2. 사용자가 입력한 할 일 추가





```

async function getMessage() {
  const myMessage = document.getElementById("chat_input").value;
  let template = `<div class="line">
    <span class="chat_box mine">${myMessage}</span></div>`;
  insetrtMessage(template)

  const options = {
    method: "POST",
    headers: {
      "Authorization": `Bearer ${API_KEY_GPT}`,
      "Content-Type": "application/json"
    },
    body: JSON.stringify({
      model: "text-davinci-003",
      prompt: myMessage,
      max_tokens: 100
    })
  }
  document.getElementById("chat_input").value = ""
  try {
    const response =
      await fetch("https://api.openai.com/v1/completions", options)
    const data = await response.json()
    let template = `<div class="line">
      <span class="chat_box">${data.choices[0].text}</span></div>`
    insetrtMessage(template)
  } catch(error) {
    console.error(error)
  }
}

function insetrtMessage(template) {
  document.getElementById("chat").insertAdjacentHTML("beforeend",
    template)
}

```

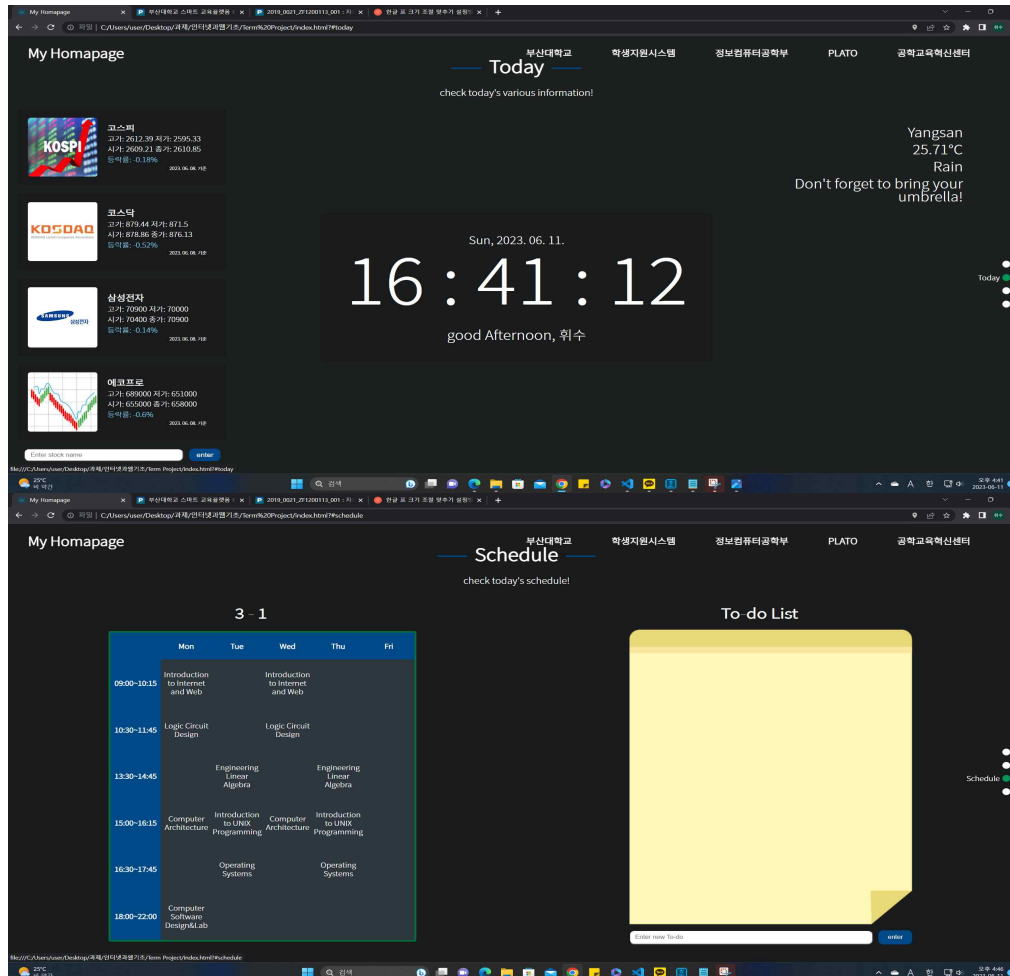
### 3. 사용자가 입력한 메시지와 반환된 메시지 출력

## I. 소스 코드에 각 함수 및 최소 개발 기준 별로 기능 설명하는 주석

소스코드 참조

**최소 개발 기준 이외에 추가 개발 항목**

## A. 웹 페이지 내부 네비게이션



홈페이지 내부 이동을 위한 우측 고정 네비게이션과 시각적 효과 구현

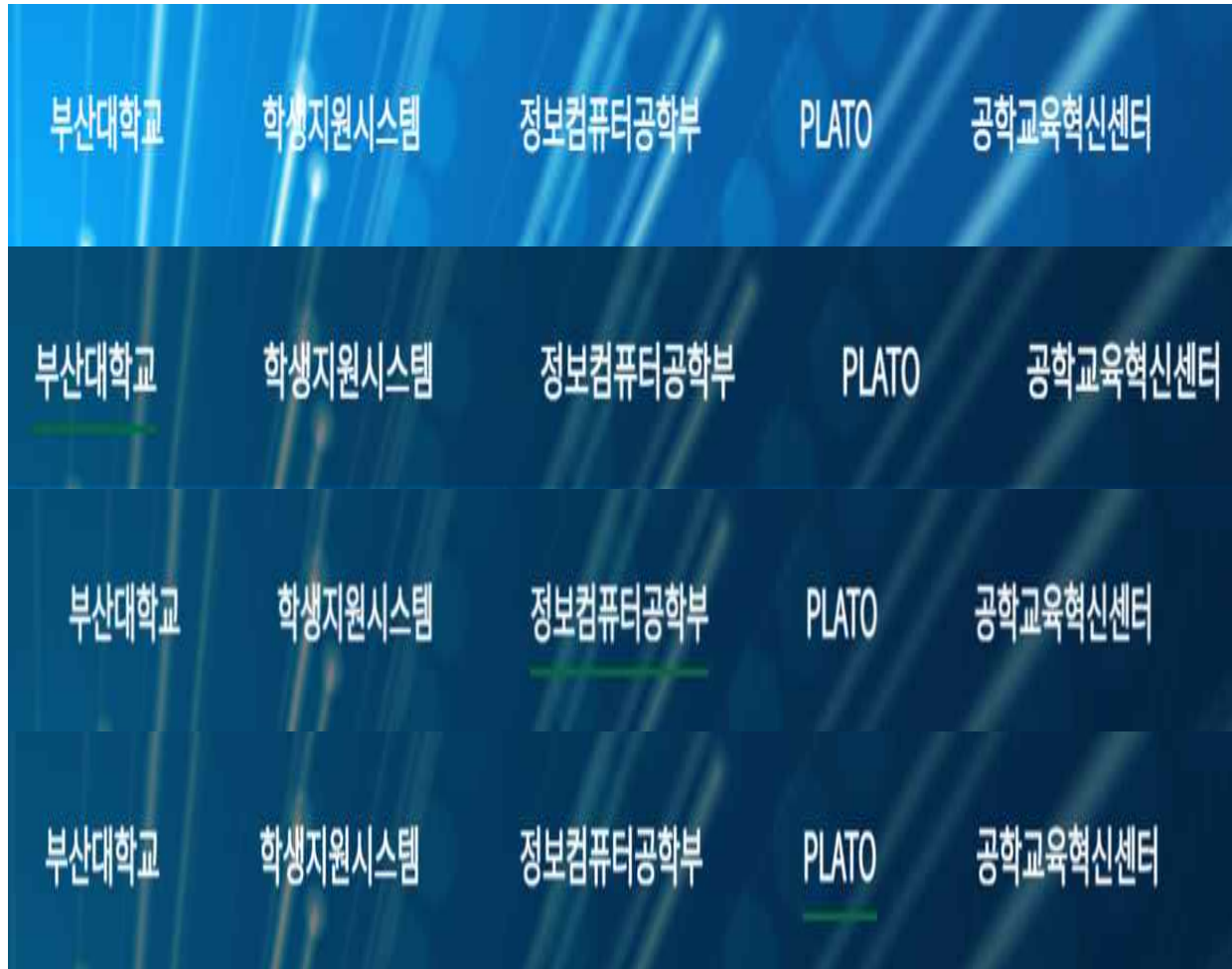
```
<div class="navigation_side_container">
  <span class="navigation_side_text">Today</span>
  <a href="#today"><div class="navigation_side_button"></div></a>
</div>
```

```
<div class="navigation_side_container">
  <span class="navigation_side_text">Schedule</span>
  <a href="#schedule"><div class="navigation_side_button"></div></a>
</div>
```

```
<div class="navigation_side_container">
  <span class="navigation_side_text">ChatGPT</span>
  <a href="#chat_gpt"><div class="navigation_side_button"></div></a>
</div>
```

```
function btnOver() {
  this.style.backgroundColor = "#029558"
}
function btnOut() {
  this.style.backgroundColor = "white"
}
```

## B. 네비게이션에 애니메이션 추가



```
#navigation_main {
  background-color: rgba( 0, 0, 0, 0 );
  transition: background-color 0.3s;
  display: flex;
  justify-content: space-between;
  padding: 0 40px;
  position: fixed;
  top: 0px;
  right: 0px;
  left: 0px;
  color: white;
}

#navigation_main:hover {
  background-color: rgba( 0, 0, 0, 0.5 );
}

#navigation_main a:hover {
  border-bottom: 3px solid #026e3c;
}

a[target = _blank] {
  border-bottom: 3px solid rgba(0, 0, 0, 0);
  transition: border-bottom 0.25s;
  padding-bottom: 5px;
}
```

마우스 이벤트에 반응하는 시각적인 네비게이션 구현

# 웹사이트 동작 시나리오

## A. 오늘의 날씨, 주가 정보 받아오기



```
async function getStockInfo(url, i) {
  await fetch(url).then(response => response.json()).then(data => {
    let stock_info = new Array(5);
    const baseDate = data.response.body.items.item[0].basDt
    const stockName = data.response.body.items.item[0].itmsNm
    ...

    stock_info[0] = `${baseDate.substr(0,4)}.
      ${baseDate.substr(4,2)}. ${baseDate.substr(6,2)}. 기준`
    ...

    document.getElementsByClassName("base_date")[i].innerHTML = stock_info[0]
    ...
  }).catch(function(error) {
    console.error(error);
  });
}

stockBtn.addEventListener("click", function() {
  const myStockName = myStock.value
  const MyUrl = `https://apis.data.go.kr/ ... itmsNm=${myStockName}`
  myStock.value = ""
  getStockInfo(MyUrl, 3)
})
```

input에 정확한 주식의 이름을 입력하고 submit하면 해당 주식의 정보를 출력한다



```
async function onGeoOk(position) {
  const lat = position.coords.latitude;
  const lon = position.coords.longitude;
  const url = `https://api.openweathermap.org/data/2.5/...`;
  await fetch(url).then(response => response.json()).then(data => {
    const city = document.getElementById("city");
    const weather = document.getElementById("weather");
    const temp = document.getElementById("temp");
    city.innerHTML = data.name;
    weather.innerHTML = data.weather[0].main;
    temp.innerHTML = `${data.main.temp}°C`;
    if(data.weather[0].main == "Rain") {
      let template = `<span class="weather_info" id="city">
        Don't forget to bring your umbrella!</span>`
      weatherInfoContainer.insertAdjacentHTML("beforeend", template)
    }
  }).catch(function(error) {
    console.error(error);
  });
}

function onGeoError() {
  alert("error");
}

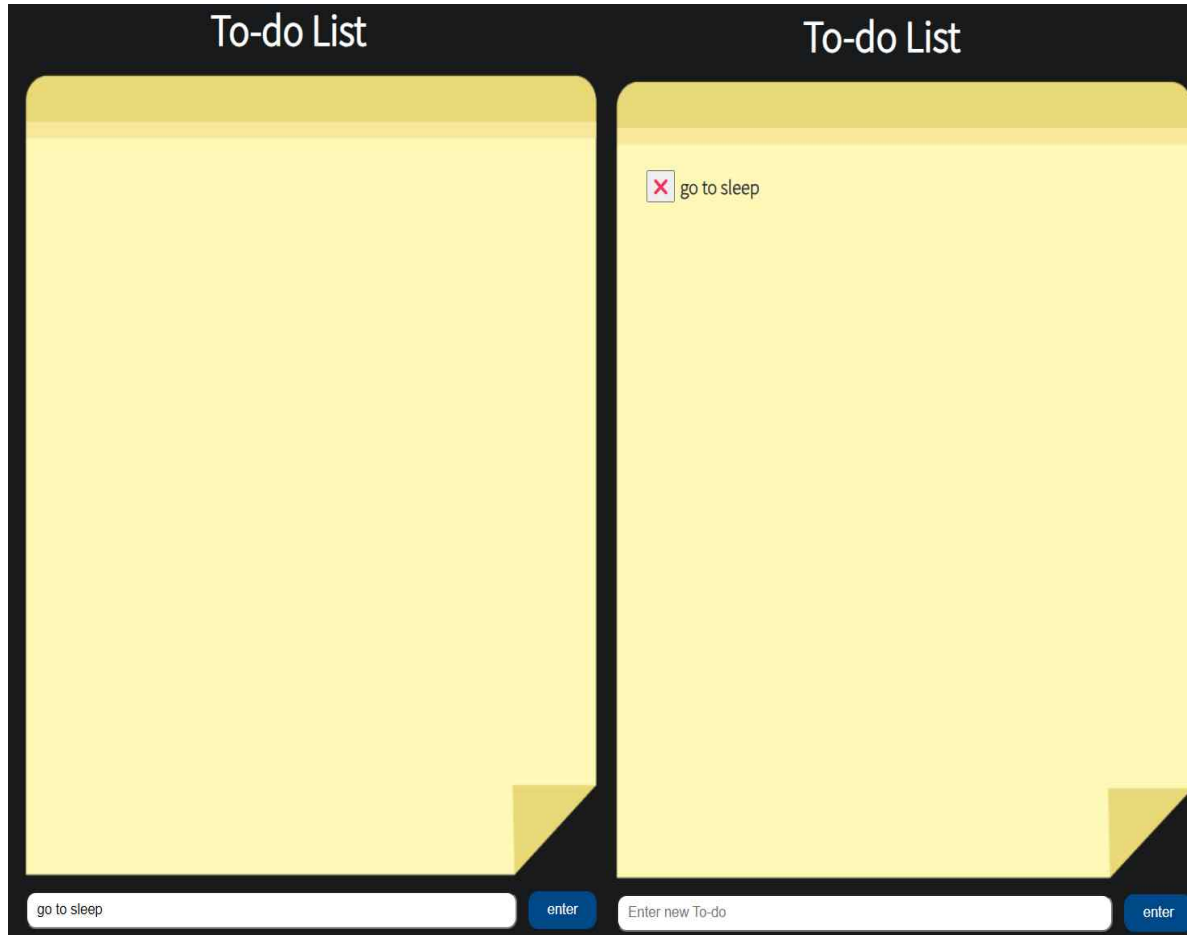
navigator.geolocation.getCurrentPosition(onGeoOk, onGeoError);
```

위치 정보를 허용하면 오늘의 날씨를 출력한다

만약 오늘이 날씨가 "Rain"이면 "Don't forget to bring your umbrella!" 문구도 출력한다



## B. 할 일 리스트 추가하기



```
function addToDo(newTodo) {
  const li = document.createElement("li")
  const span = document.createElement("span")
  const button = document.createElement("button")
  li.setAttribute("class", "todo_li")
  span.setAttribute("class", "todo_span")
  button.setAttribute("class", "todo_button")
  span.innerText = newTodo
  button.innerText = "X"
  button.addEventListener("click", removeToDo)
  li.appendChild(button)
  li.appendChild(span)
  toDoList.appendChild(li)
}

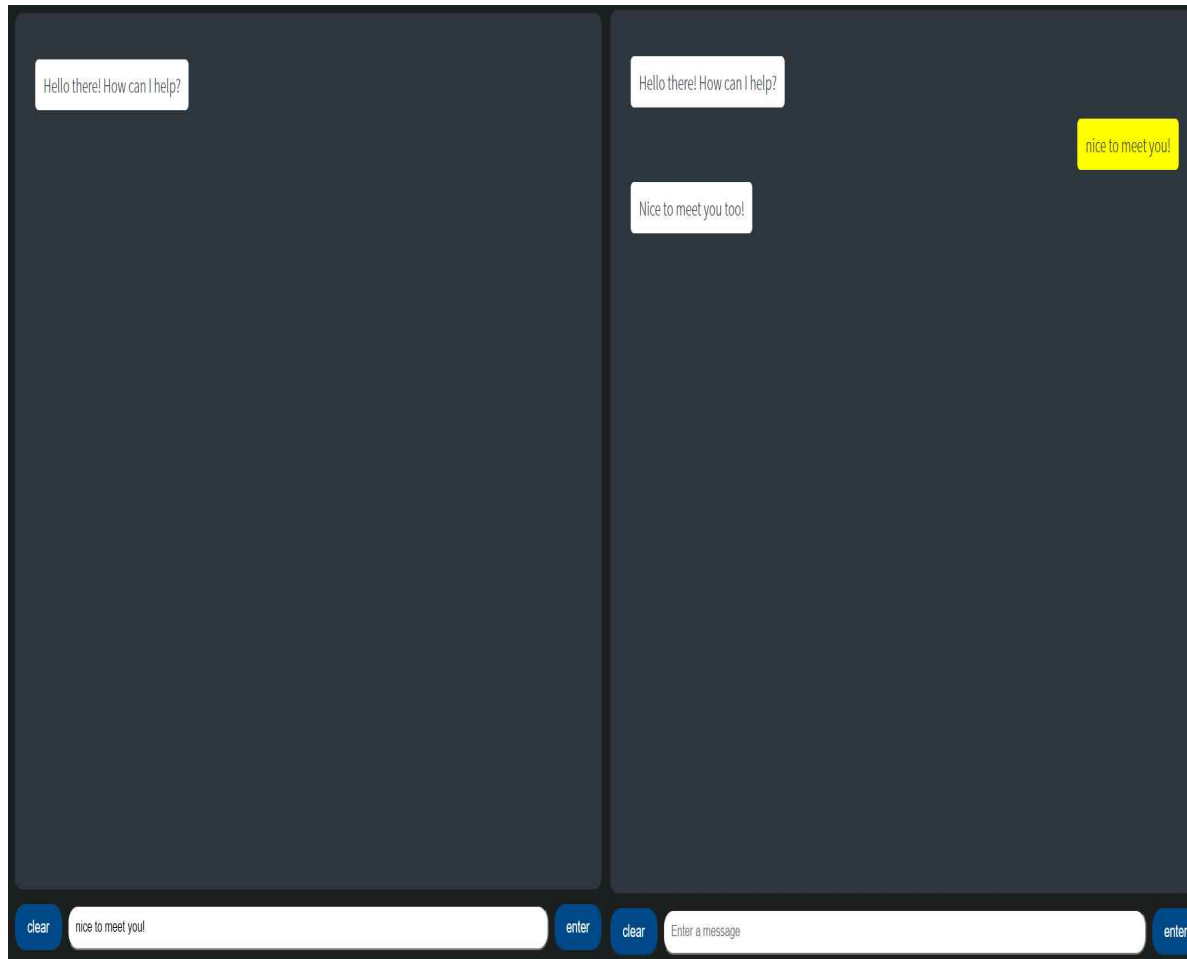
function removeToDo(event) {
  const li = event.target.parentElement
  li.remove()
}

function toDoSubmit() {
  const newTodo = toDoInput.value
  if(newTodo !== "") {
    toDoInput.value = ""
    addToDo(newTodo)
  }
}
```

input에 새로운 할 일을 입력하고 submit하면 할 일 리스트에 추가된다.

X 버튼을 눌러 완료된 할 일을 삭제할 수 있다

## C. ChatGPT 기반 챗봇과 대화하기



```
async function getMessage() {
  const myMessage = document.getElementById("chat_input").value;
  let template = `<div class="line">
    <span class="chat_box mine">${myMessage}</span></div>`;
  insetrtMessage(template);

  const options = {
    method: "POST",
    headers: {
      "Authorization": `Bearer ${API_KEY_GPT}`,
      "Content-Type": "application/json"
    },
    body: JSON.stringify({
      model: "text-davinci-003",
      prompt: myMessage,
      max_tokens: 100
    })
  };
  document.getElementById("chat_input").value = "";
  try {
    const response =
      await fetch("https://api.openai.com/v1/completions", options);
    const data = await response.json();
    let template = `<div class="line">
      <span class="chat_box">${data.choices[0].text}</span></div>`;
    insetrtMessage(template);
  } catch (error) {
    console.error(error);
  }
}

function insetrtMessage(template) {
  document.getElementById("chat").insertAdjacentHTML("beforeend",
    template);
}
```

Input에 보내고 싶은 메시지를 입력하고 submit하면 내 메시지와 답장을 출력한다  
clear 버튼을 눌러 메시지를 삭제할 수 있다

**보완해야 할 부분**

## A. 화면 크기에 따른 깨짐 현상

이 프로젝트 내에서는 웹페이지를 해상도가 2560 \* 1440 의 환경을 기준으로 작성하였다.  
하지만 해상도가 다른 환경에서 layout이 깨지는 상황이 발생하였다. 이는 상대적 단위인 %, vh, vw 등을 활용하지 않고 px 등의 절대적 단위를 이용하여 css파일을 작성하였기 때문이다. 작업 환경이 아닌 다른 환경에서도 잘 출력이 되게 하려면 상대적 단위만을 사용하여 작성하는 것이 바람직해 보인다.

## B. 화면의 구성과 id와 class의 통일성

html코드를 작성하기 이전에 어느 contents들을 div로 묶고 활용할 것인지 미리 고려해야 하는 게 좋았을 것이라고 느꼈다.  
큰 그림을 먼저 그리고 그 안에 어떤 contents들을 어떻게 배치할 것인지를 먼저 구상해야 한다.  
그리고 id와 class를 일정한 규칙을 세워서 부여하고 이름을 짓는다면 작업하기 훨씬 편리할 것이다.  
나중에 다시 소스코드를 보더라도 알아볼 수 있도록 이름을 짓는 방법론이 필요할 것이다.

## C. 다양한 API 요청 방법

이 프로젝트 내에서는 fetch 함수만을 이용해서 API 요청을 했지만 더욱 다양한 요청 방법을 찾을 수 있었다.  
ajax, node.js 등 여러 방법을 공부하고 활용하면 좋을 것이다.