

1

# 가로 웹 페이지

## 반응형 웹사이트

URL

<https://youtu.be/5nGJwDg2J08>

특징

가로로 스크롤이 되는 웹 페이지

구현언어

HTML, CSS, JavaScript



## Main Page

### HTML

BEM 방식을 사용

### Css

가로 형 스크롤 페이지를 구현

### JavaScript

Swiper Api 를 사용하여 side Scroll이 가능하게 함

### nav

클릭 시 숨겨진 메뉴 창이 나오  
게 된다.



### Section

웹 페이지에 대한 정보를 제공  
하는 자동슬라이더와 Scroll이  
가능한 홈페이지

# Sub Page

HTML

BEM 방식을 사용

Css

가로 형 스크롤 페이지를 구현

JavaScript



Section

Section

Section

Section

Footer

회원가입

팀에 대한 정보  
확인

팀 랭킹 확인

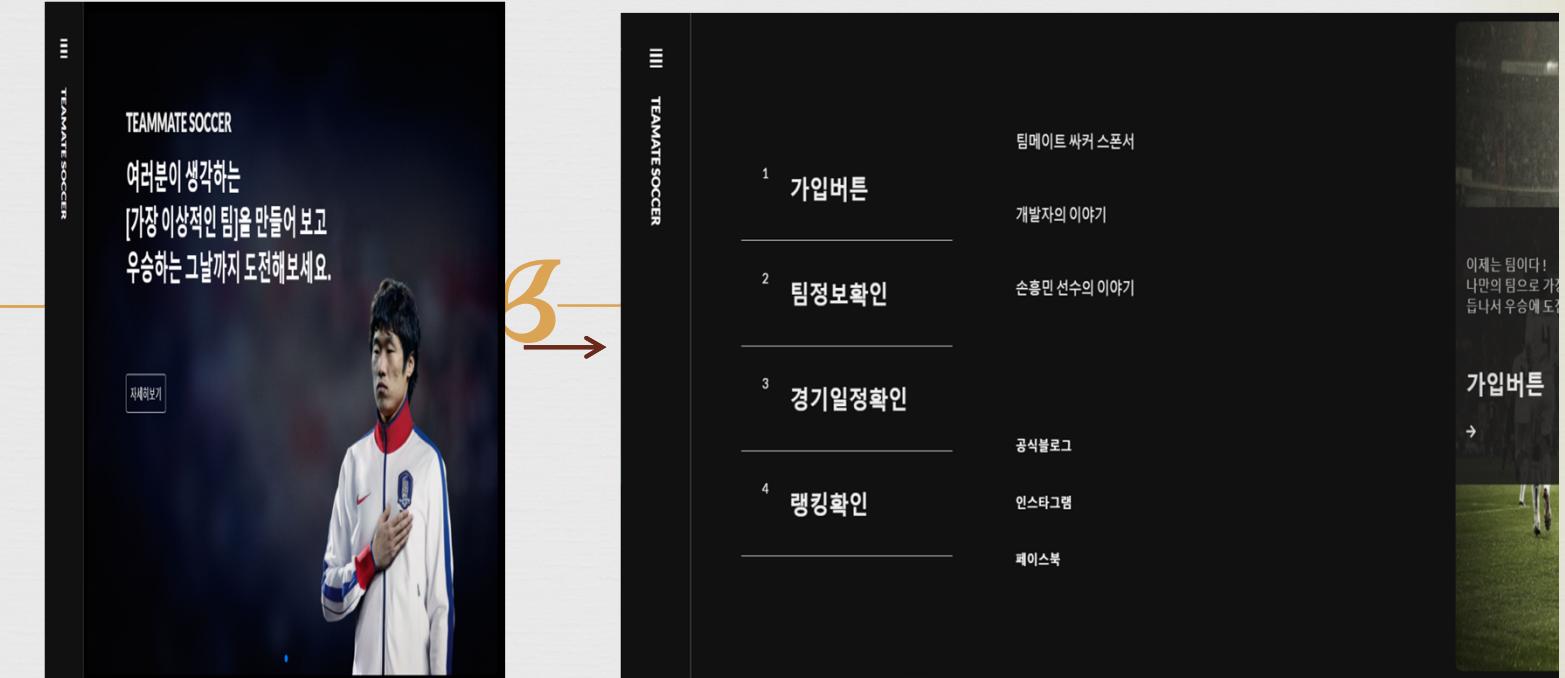
경기 일정 확인

부가적인 내용

# function

## 1. 메뉴 창

Swiper api를 사용하여  
side scroll 기능이 가능하게  
함



```
// side navbar

const menu__container = document.querySelector(".menu__container");
const menu__btn = document.querySelector(".menu__button");
const intro = document.querySelector("#intro");

menu__btn.addEventListener("click", () => {
    menu__container.classList.toggle("show_navbar");

    if (intro.style.zIndex == -1) {
        intro.style.zIndex = 0;
    } else {
        intro.style.zIndex = -1;
    }
});
```

2

# Yoga

## 모바일 웹사이트

URL

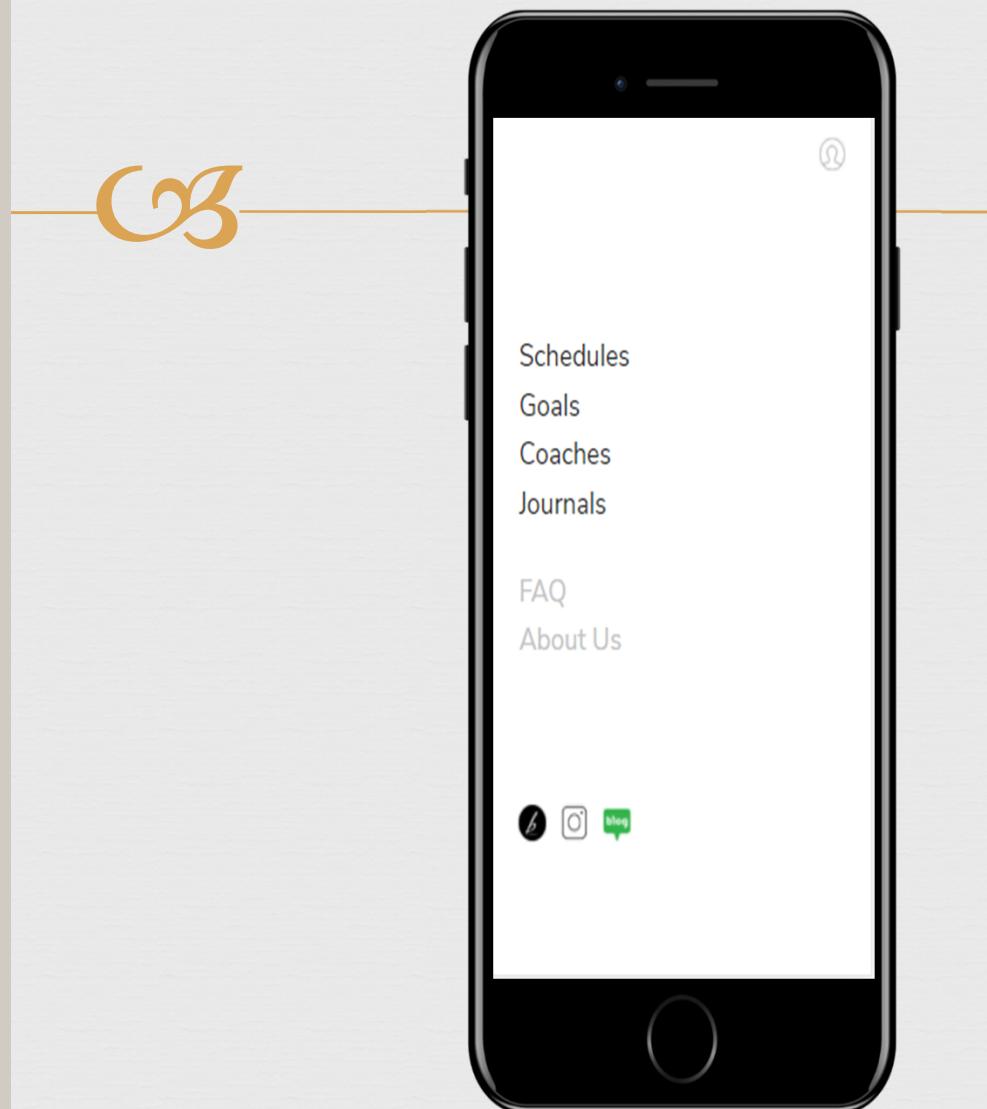
<https://youtu.be/R1IuXyHbLLc>

특징

모바일 웹 기반 Layout 구현

구현언어

HTML, CSS, JavaScript



# Main Page

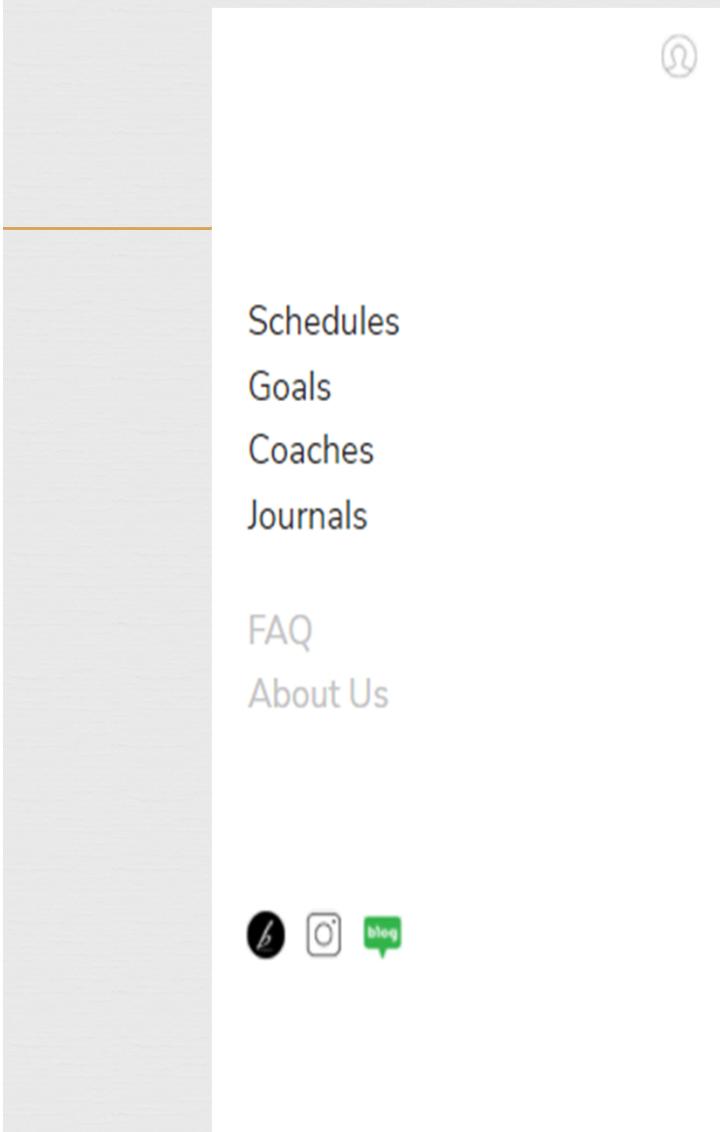
**HTML**

BEM 방식을 사용

**Css**

Mobile web 환경에 맞게 %  
를 사용함

**JavaScript**



**Section**



**Header**



**Main**

메뉴, 각 페이지로 이동하는 링크버튼 포함



**Footer**

SNS로 이동할 수 있도록 구현

## Sub Page

### HTML

BEM 방식을 사용

### Css

Flex를 활용한 정렬, 자주 사용될 CSS요소들은 root를 만들어 사용

### JavaScript

1. Swiper Api 를 사용하여 side Scroll이 가능하게 함
2. 데이터를 동적으로 가져와서 HTML에 값을 추가



### Section

### Header

메인 페이지로 이동하는 링크 버튼

### Main

요일마다 요가 일정을 보여주는 일정표

클릭 시 상세페이지(session) 가 나오게끔 링크버튼 포함

사이드로 스크롤이 가능하며, 현재 날짜에 따라 스케줄이 내용이 보여짐

## Sub Page

### HTML

BEM 방식을 사용

### Css

Flex를 활용한 정렬, 자주 사용될 CSS요소들은 root를만 들어 사용

### JavaScript

The screenshot shows a mobile application interface. At the top, there is a navigation bar with a back arrow, the text "Session", and a user profile icon. The main content area features a large image of a woman performing a complex yoga pose (Lord of the Dance). Below the image, the title "새벽을 깨우는 요가" (Morning Yoga) is displayed, followed by the subtitle "#필лекс요가 #목척추글반". A descriptive paragraph in Korean follows, mentioning the benefits of morning yoga. At the bottom of the screen, there is a section titled "세션구성" (Session Configuration) which includes a small thumbnail image of a schedule grid.

### Section

### Header

스케줄 페이지로 이동하는 링크 버튼

### Main

Schedules 페이지의 상세 페이지로 각 요 스케줄 일정과 설명

# Sub Page

## HTML

BEM 방식을 사용

## Css

Flex를 활용한 정렬, 자주 사용될 CSS요소들은 root를 만들어 사용

## JavaScript

1. Bootstrap API 활용

2. Switch(past, current)  
가 변화할 때마다 값이 바뀌는  
기능 구현



### Section

### Header

메인 페이지로 이동하는 링크 버튼

### Main

현재와 과거의 목표와 목표달성을  
여부를 보여줌

Switch 이동을 통해 현재의 목표와 과거의 목표를 확인 가능

### Footer

현재 페이지에 대한 설명,  
Switch가 past일 때는 사라짐

## Sub Page

### HTML

BEM 방식을 사용

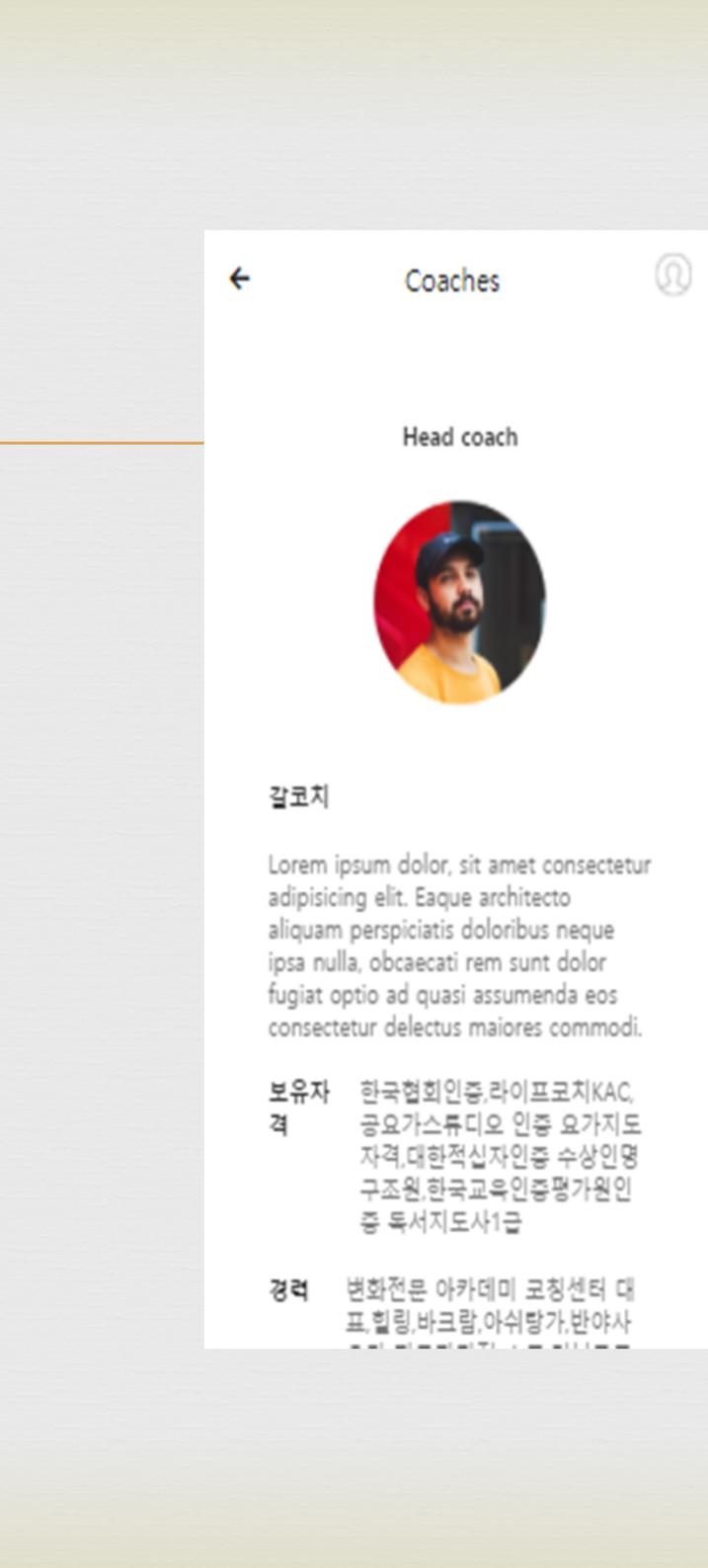
### Css

Flex를 활용한 정렬, 자주 사용될 CSS요소들은 root를 만들어 사용

### JavaScript

1. Swiper Api 를 사용하여 side Scroll이 가능하게 함

2. 데이터를 동적으로 가져와서 HTML에 값을 추가



### Section

### Header

메인 페이지로 이동하는 링크 버튼

### Main

사이드로 스크롤이 가능하며, 보유자격 및 경력 등 코치 별 정보가 나타남

## Sub Page

HTML

BEM 방식을 사용

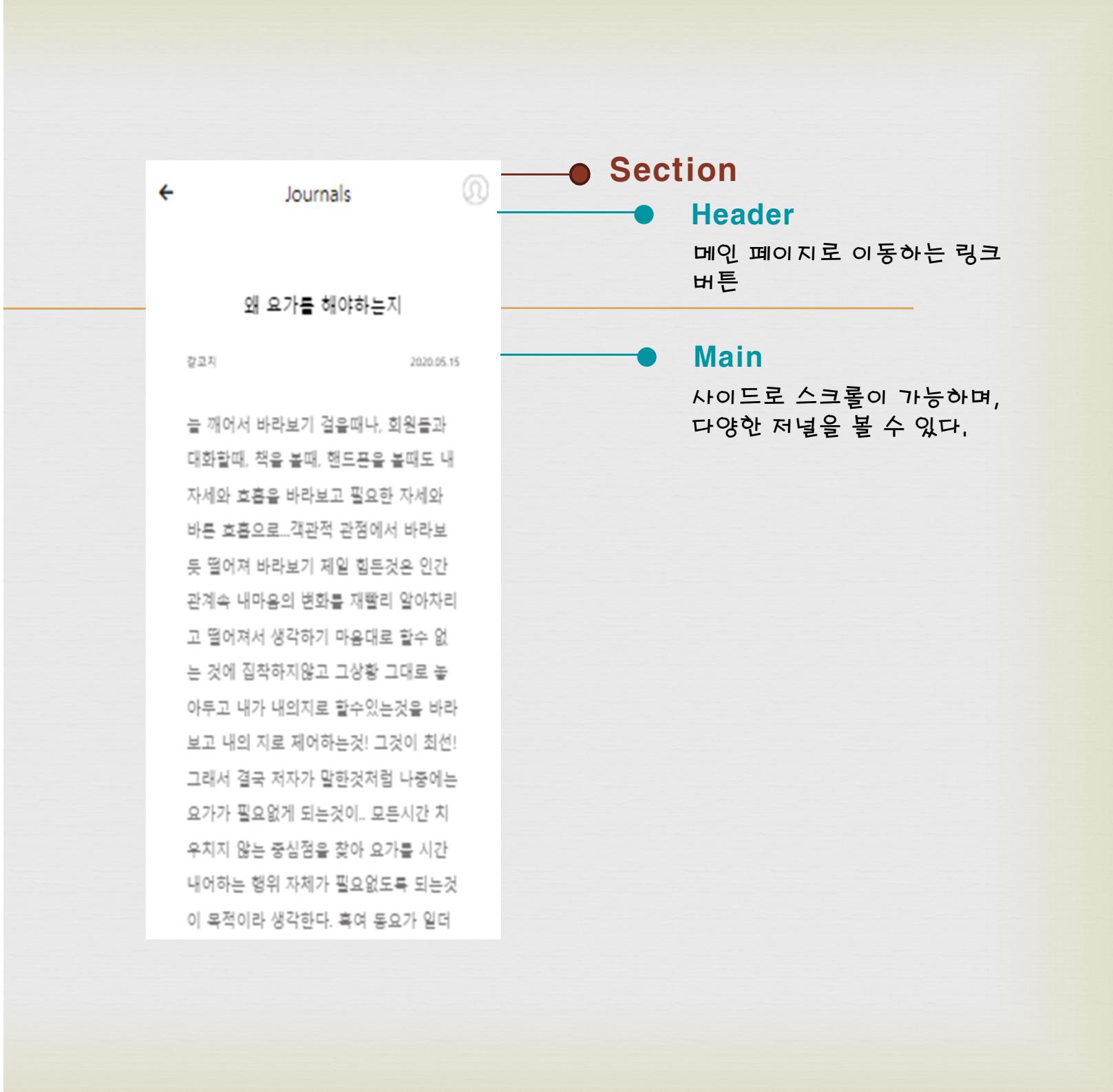
Css

Flex를 활용한 정렬

JavaScript

1. Swiper Api 를 사용하여  
side Scroll이 가능하게 함

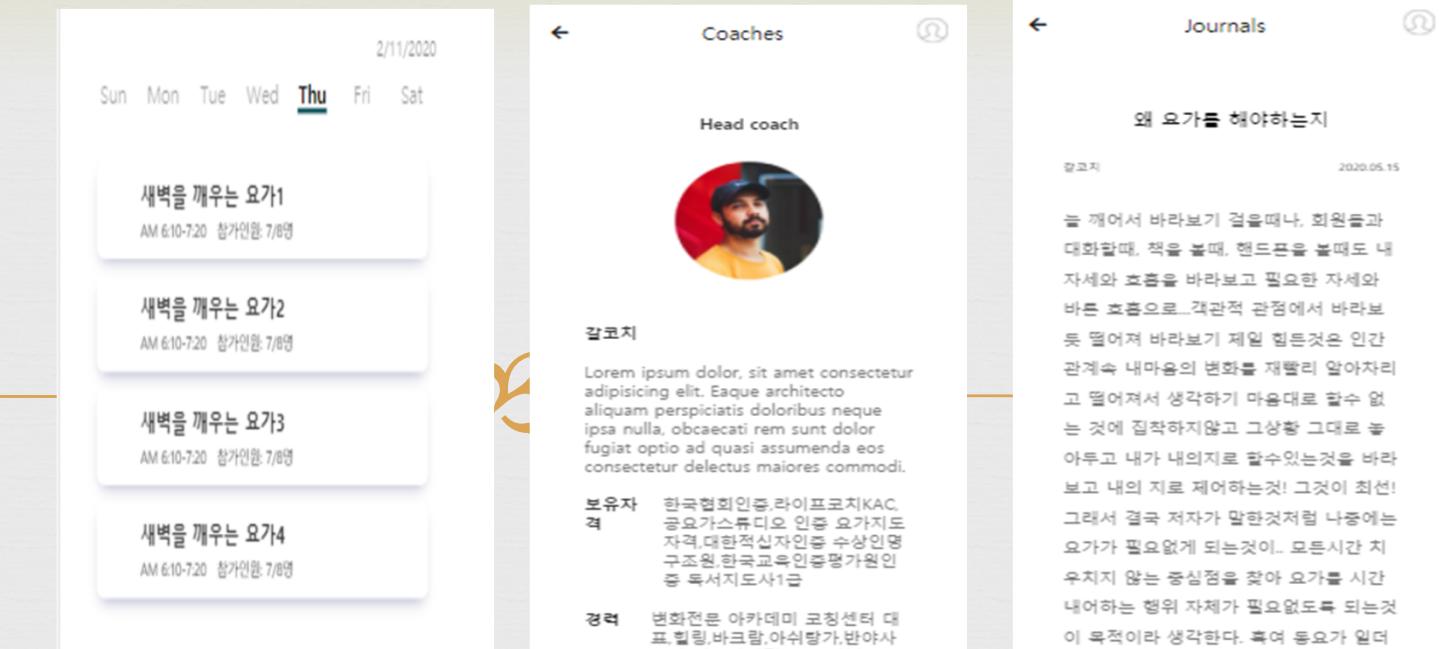
2. 데이터를 동적으로 가져와  
서 HTML에 값을 추가



# function

## 1. Side Scroll

Swiper api를 사용하여  
side scroll 기능이 가능하게  
함



```
// swiper
let day;
const swiper = new Swiper(".swiper-container", {
  pagination: {
    el: ".schedules_date ul",
    clickable: true,
    renderBullet: function (index, className) {
      switch (index) {
        case 0:
          day = "Sunday";
          break;
        case 1:
          day = "Monday";
          break;
        case 2:
          day = "Tuesday";
          break;
        case 3:
          day = "Wednesday";
          break;
        case 4:
          day = "Thursday";
          break;
        case 5:
          day = "Friday";
          break;
        case 6:
          day = "Saturday";
          break;
      }
      return `li class=${className} name=${day}>${day.slice(0, 3)}</li>`;
    },
  },
});
```

```
var swiper = new Swiper(".journals_main", {
  navigation: {
    nextEl: ".swiper-button-next",
    prevEl: ".swiper-button-prev",
  },
});
```

```
// slider
var swiper = new Swiper(".coaches_main", {
  navigation: {
    nextEl: ".swiper-button-next",
    prevEl: ".swiper-button-prev",
  },
});
```

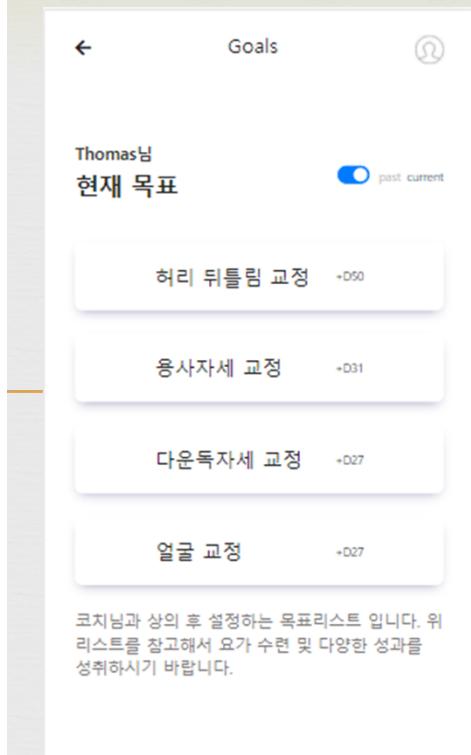
# function

## 1. 동적인 데이터를 가져와 HTML에 추가하여 View 갱신

데이터 추가 및 삭제 등을 할 때 HTML을 건들지 않기 위해 구현

## 2. Switching에 따른 값 변화

Switch를 바꿀 때마다 보여지는 내용이 다르게 구현



### 1. 동적인 데이터

```
let objects = [
  one: {
    state: "current",
    title: "하리 뒤틀림 교정",
    date: "+D50",
  },
  two: {
    state: "past",
    title: "코브라 자세",
    date: "complete",
  },
  three: {
    state: "past",
    title: "거북목 교정",
    date: "complete",
  },
  four: {
    state: "current",
    title: "용사자세 교정",
    date: "+D31",
  },
  five: {
    state: "current",
    title: "다운독자세 교정",
    date: "+D27",
  },
  six: {
    state: "current",
    title: "얼굴 교정",
    date: "+D27",
  },
];
```

```
// switch change
// switch를 눌렀을 때 현재에 있던 것들이 과거의 값으로 바뀌는 함수
const switch_btn = document.querySelector(".switch_btn");
const switch_name = document.querySelectorAll(".switch_name");

const switch_title = document.querySelectorAll(".switch_title");
const lists = document.querySelectorAll(".list");

const main_footer = document.querySelector(".goals_footer");

switch_btn.addEventListener("click", () => {
  switch_name[0].classList.toggle("switch_open");
  switch_name[1].classList.toggle("switch_open");
  switch_title[0].classList.toggle("obj_open");
  switch_title[1].classList.toggle("obj_open");

  lists.forEach((list) => {
    list.classList.toggle("list_open");
  });

  main_footer.classList.toggle("footer_hide");
});
```

### 2. 스위치를 바꿀 때마다 보여지는 내용을 다르게 구현

```
document.querySelector(".goals_main");
const main_list = document.querySelector(".main_list");
let temp_str = "";

Object.keys(objects).forEach((key) => {
  if (objects[key].state == "current") {
    const titleNode = `<h3 class="list_title">${objects[key].title}</h3>`;
    const dateNode = `<span class="list_date">${objects[key].date}</span>`;

    temp_str += `<li class="list"><div class="list_box">${titleNode}${dateNode}</div></li>`;
  } else if (objects[key].state == "past") {
    const titleNode = `<h3 class="list_title">${objects[key].title}</h3>`;
    const dateNode = `<span class="list_date">${objects[key].date}</span>`;

    temp_str += `<li class="list list_open"><div class="list_box">${titleNode}${dateNode}</div></li>`;
  }
}

main_list.innerHTML = temp_str;
```

3

# 포트폴리오

반응형 웹사이트

URL

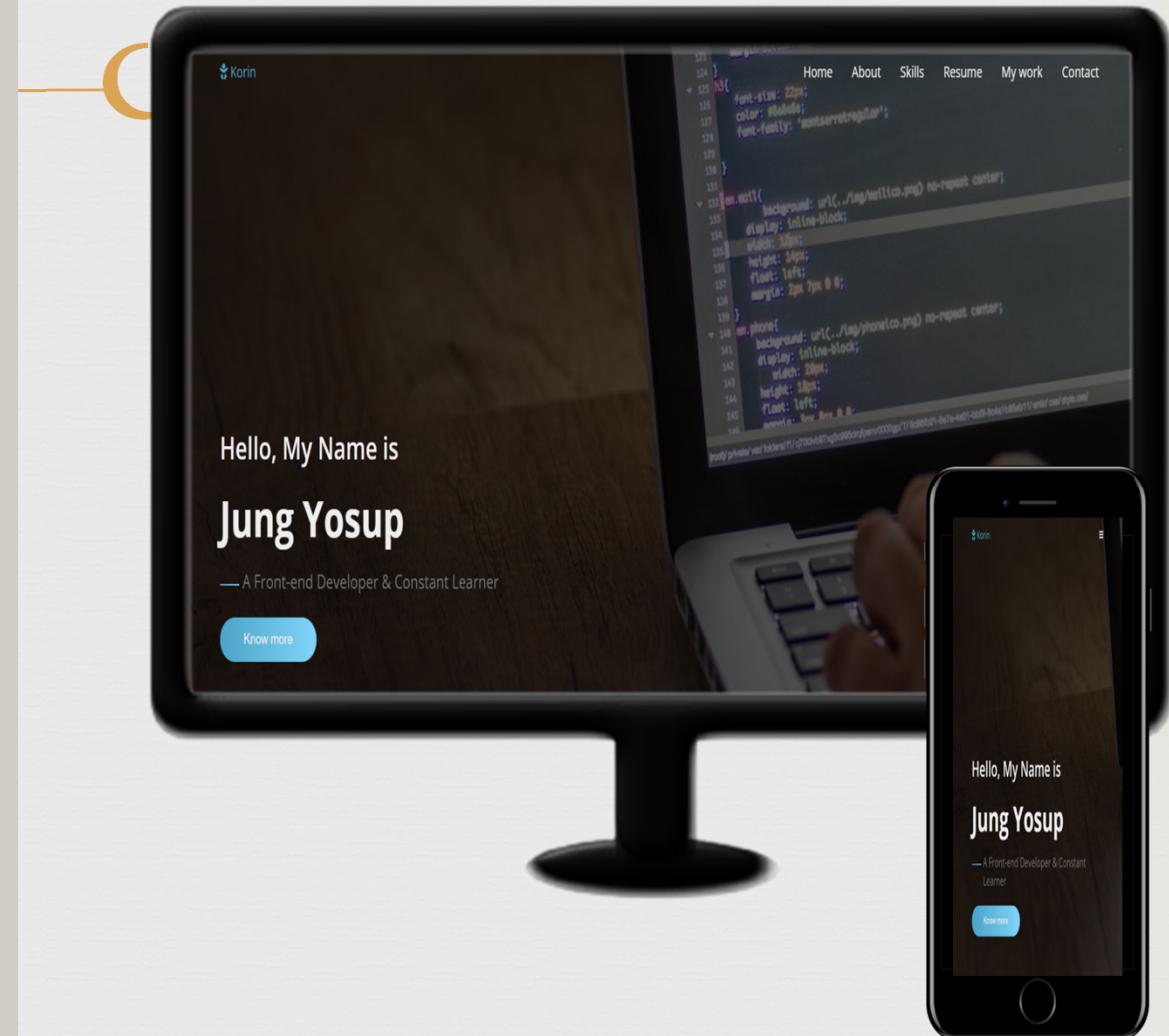
<https://youtu.be/Ce3XXm-XtGk>

특징

반응형 웹 사이트

구현언어

HTML, CSS, JavaScript



# Main Page

HTML

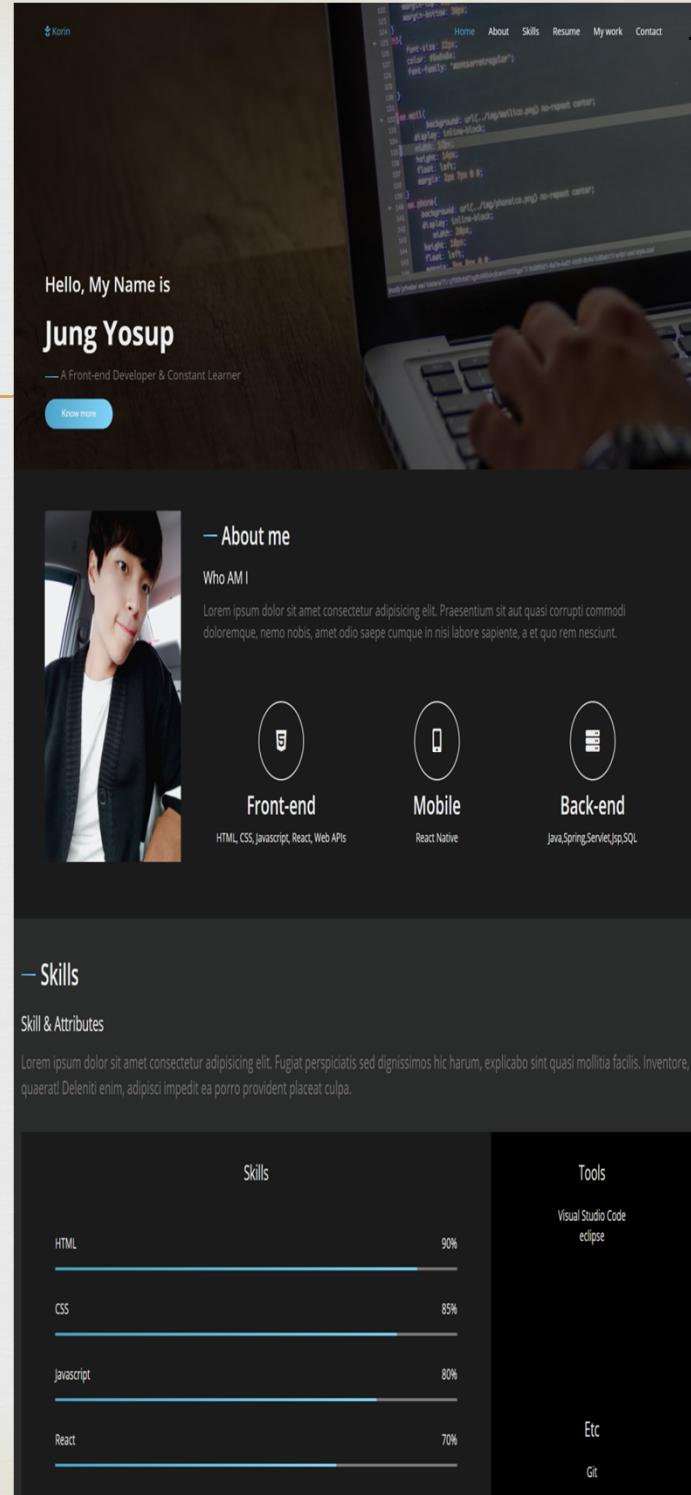
BEM 방식을 사용

Css

Flex를 활용한 정렬

JavaScript

Scroll이 skills으로 이동할 때  
skill의 퍼센트가 증가하는 애니  
메이션 구현



## nav

버튼 클릭 시 각 Section으로 이동하는 기능 구현  
position : fixed를 활용한 navbar 고정

모바일에서는 bars 형태로 나타남

## Section1: home

## Section2 : about me

About me, Jung Yosup의 대해 알려주는 부분

## Section3 : skills

어떤 기술을 가지고 있고 실력이 어떤지 나타내는 부분

Scroll이 해당 section으로 이동할 때 skill의 퍼센트가 증가하는 애니메이션 구현

# Main Page

## HTML

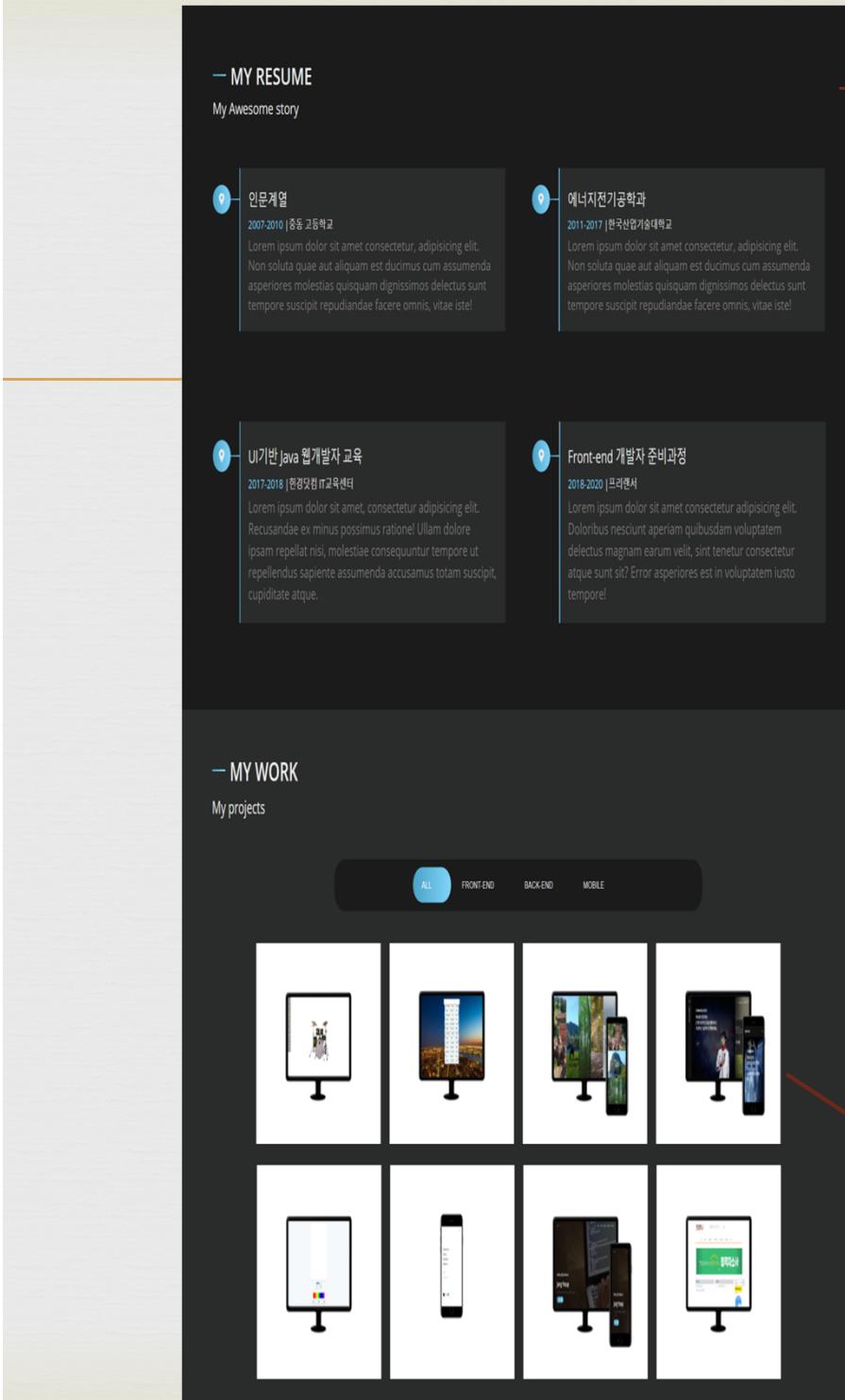
BEM 방식을 사용

## Css

Flex를 활용한 정렬

## JavaScript

데이터를 동적으로 가져와  
HTML에 값을 추가



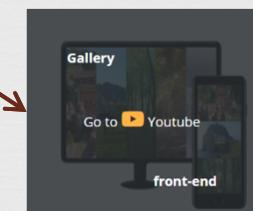
## Section4: My Resume

메뉴, 각 페이지로 이동하는 링크버튼 포함

## Section5: My Resume

카테고리를 클릭하면 해당 카테고리에 따라 프로젝트들이 분류됨

카테고리를 클릭하면 카테고리에 따른 프로젝트들의 개수를 확인할 수 있음



이미지 오버시  
YouTube로 이동할 수 있는 화면으로 전환

# Main Page

## HTML

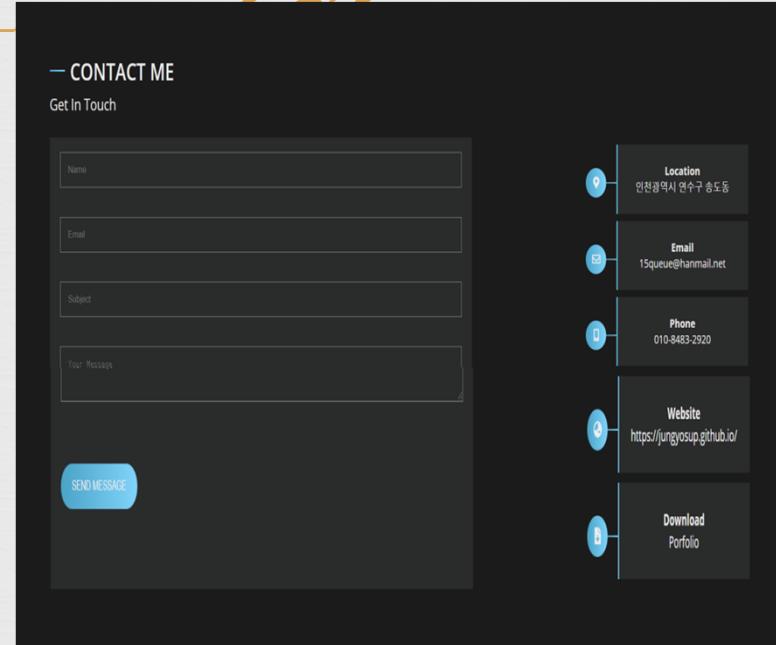
BEM 방식을 사용

## Css

Flex를 활용한 정렬

## JavaScript

Emailjs Api를 사용하였다.



## Section5: Contact Me

깃 허브 블로그와 포트폴리오를  
다운받을 수 있는 링크구현

나에게 이 메일을 보낼 수 있게  
기능 구현

# Responsive Page

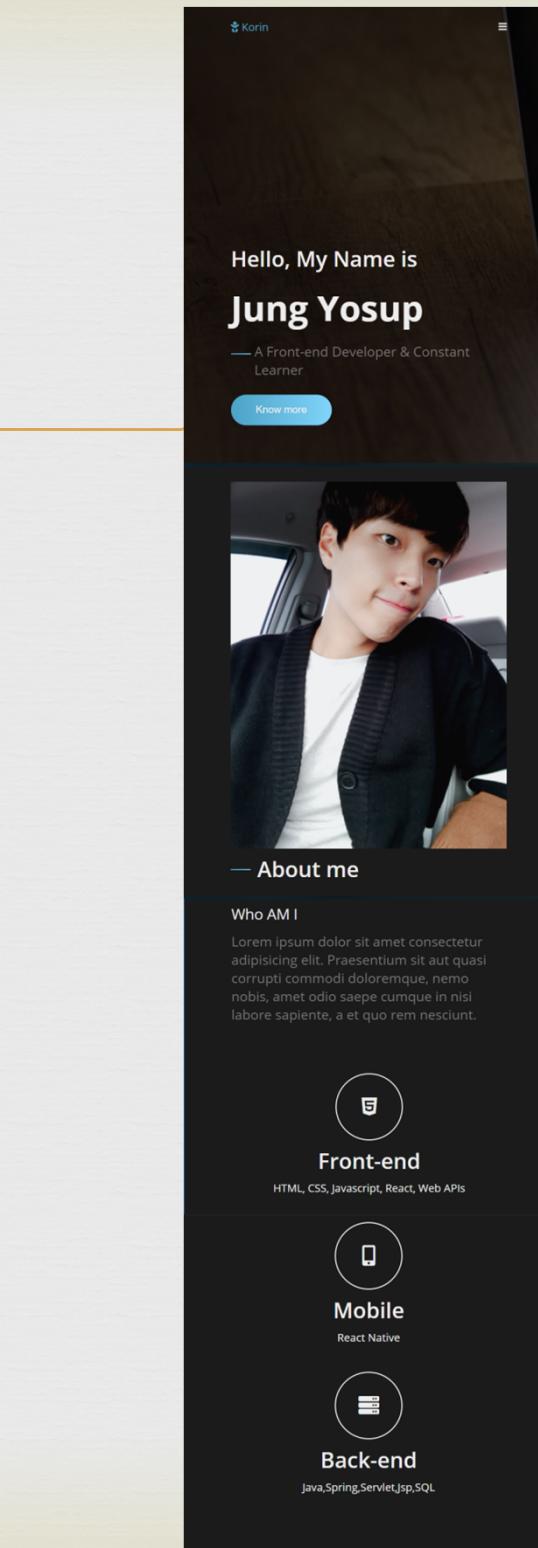
## HTML

BEM 방식을 사용

## Css

미디어 쿼리를 사용했고, 미디어 쿼리 안에서는 최소한의 CSS 작업  
(760px 이하)

## JavaScript



**Skills**

**Skill & Attributes**

Lorem ipsum dolor sit amet consectetur adipisicing elit. Fugiat perspiciatis sed dignissimos hic harum, explicabo sint quasi mollitia facilis. Inventore, quae! Deleniti enim, adipisci impedit ea porro provident placeat culpa.

Skills	Progress (%)
HTML	90%
CSS	85%
Javascript	80%
React	70%
Java	80%

**Tools**

Visual Studio Code, eclipse

**Etc**

Git

**MY RESUME**

My Awesome story

**인문계열**  
2007-2010 | 중등 고등학교  
Lorem ipsum dolor sit amet consectetur, adipisicing elit. Non soluta quae aut aliquam est ducimus cum assumenda asperiores molestias quisquam dignissimos delectus sunt tempore suscipit repudiandae facere omnis, vitae iste!

**인문계열**  
2007-2010 | 중등 고등학교  
Lorem ipsum dolor sit amet consectetur, adipisicing elit. Non soluta quae aut aliquam est ducimus cum assumenda asperiores molestias quisquam dignissimos delectus sunt tempore suscipit repudiandae facere omnis, vitae iste!

**MY WORK**

My projects

**ALL FRONT-END BACK-END MOBILE**

**CONTACT ME**

Get In Touch

Name:

Email:

Subject:

Your Message:

SEND MESSAGE

**Location**  
인천광역시 연수구 송도동

**Email**  
15queue@hanmail.net

**Phone**  
010-8483-2920

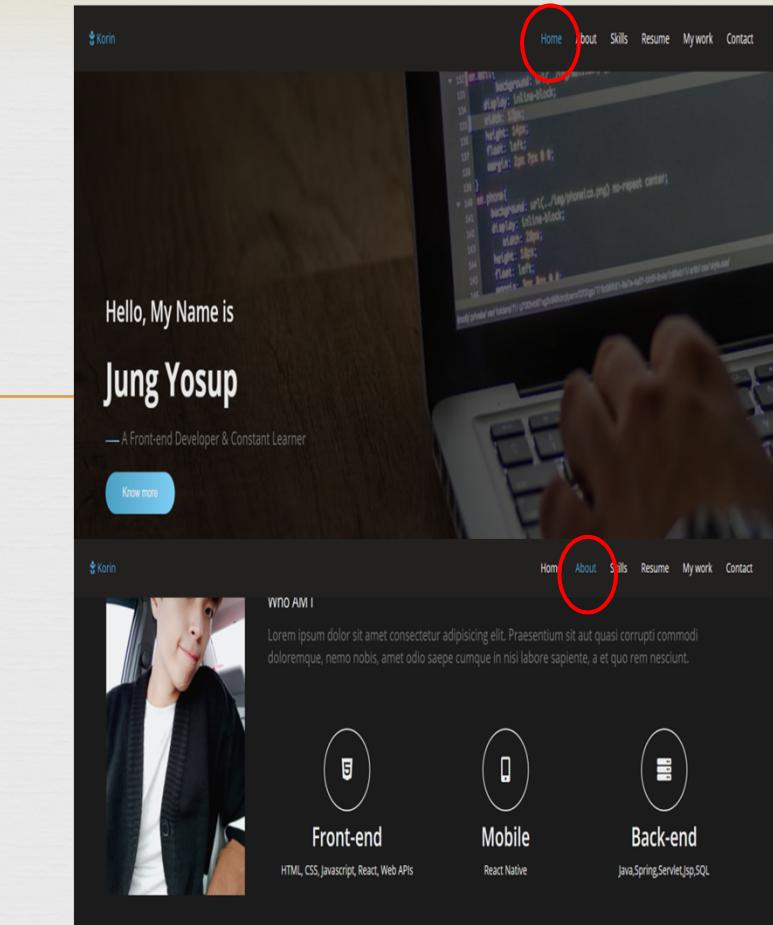
**Website**  
<https://jungyosup.github.io/>

**Download**  
Portfolio

# function

## 1. Section 위치에 따른 Navbar 버튼 색상 변경

화면의 Scroll 시 위치 값과 각 Section의 높이를 구하여서 조건문을 통해 버튼 색상을 변경하였다.



```
// 3. getHeight function
const getHeight = (section) => {
  return section.getBoundingClientRect().height;
};

// 4. getPrevSectionHeight function
// 재귀 함수 사용
// section을 넣으면 자기 section 전 section까지 높이를 더해서 구해줌
const getPrevHeight = (section) => {
  const prevSection = section.previousElementSibling;
  if (prevSection == null) {
    return getHeight(section);
  } else {
    return getHeight(section) + getPrevHeight(prevSection);
  }
};
```

```
// 5. Button lights up when scrolling
const scrollLightButton = () => {
  const list = document.querySelectorAll(".navbar_list");
  document.addEventListener("scroll", () => {
    const total_prevSectionOne = getPrevHeight(navbar);
    const total_prevSectionTwo = getPrevHeight(home);
    const total_prevSectionThree = getPrevHeight(about);
    const total_prevSectionFour = getPrevHeight(skills);
    const total_prevSectionFive = getPrevHeight(resume);
    const total_prevSectionSix = getPrevHeight(work);
    const total_prevSectionSeven = getPrevHeight(contact);

    const a = window.scrollY * 1.01;

    if (total_prevSectionOne < a && a < total_prevSectionTwo) {
      addSelected(list[0]);
    } else if (
      total_prevSectionTwo < window.scrollY &&
      window.scrollY < total_prevSectionThree
    ) {
      addSelected(list[1]);
    } else if (total_prevSectionThree < a && a < total_prevSectionFour) {
      addSelected(list[2]);
      setSkillValue();
    } else if (total_prevSectionFour < a && a < total_prevSectionFive) {
      addSelected(list[3]);
    } else if (total_prevSectionFive < a && a < total_prevSectionSix) {
      addSelected(list[4]);
    } else if (total_prevSectionSix < a) {
      addSelected(list[5]);
    }
  });
};
```

3. Scroll 위치에 따른  
값과 section의 높이를  
비교하여 버튼의 색상을  
입힘

1. 각 section의  
높이를 구하는  
함수

2. 재귀 함수를 현재  
section과 이전의 높  
이의 합을 구하는 함  
수

# function

## 2. Scroll위치에 따른 애니메이션 구현

Skills section 위치로 scroll이 이동하면 스킬 값이 증가한다.

### Skills

HTML	90%
CSS	85%
Javascript	80%
React	70%
Java	80%

### Skills

HTML	90%
CSS	85%
Javascript	80%
React	70%
Java	80%

```
// 12. animation skillvalue

const setSkillValue = () => {
  const skill_values = document.querySelectorAll(".skill_value");
  let count = 0;
  Object.keys.skills_obj.forEach((key) => {
    const skill = skills_obj[key];
    const skill_experience = `${skill.ex}`;

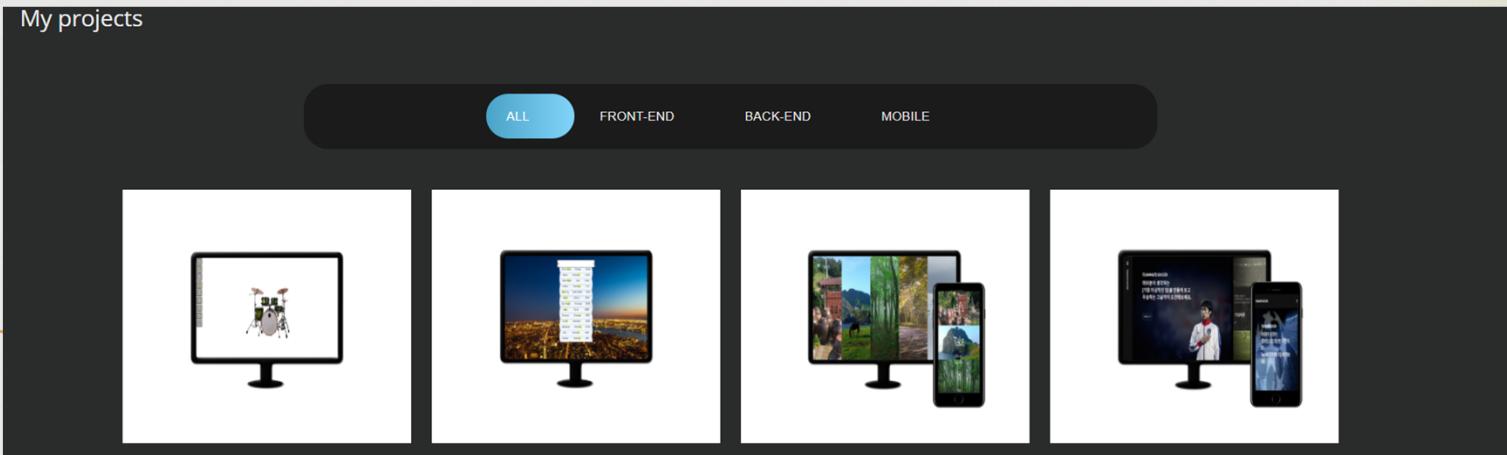
    skill_values[count].style.width = skill_experience;
    count += 1;
  });
};
```

[Home](#) [About](#) [Skills](#) [Resume](#) [My work](#) [Contact](#)

# function

## 3. Project Filtering

Data- 속성을 이용하여  
Json 데이터를 filtering 함



1. JSON 형태의 데이터를  
받아 HTML에 나타나게  
함

```
// 9. Projects Scattering to HTML
const scattering_Project = () => {
  const projectContainer = document.querySelector(".work_projects");

  let temp_str = "";
  Object.keys(projects_obj).forEach((key) => {
    const temp_obj = projects_obj[key];
    const name = temp_obj.name;
    const url = temp_obj.url;
    const part = temp_obj.part;
    const img = temp_obj.img;

    temp_str += `<a href="${url}" class="project" data-type="${part}>
      
      <div class="project_description">
        <h4 class="project_title">${name}</h4>
        Go to
        <i class="fab fa-youtube project_icon"></i>
        Youtube
        <h4 class="project_part">${part}</h4>
      </div>
    </a>`;
  });
}
```

2. data- 를 속성을 이  
용하여 데이터들을  
filtering 함

```
// 11. project filtering
const filtering_Project = () => {
  const workBtnContainer = document.querySelector(".work_categories");
  const projectContainer = document.querySelector(".work_projects");
  const projects = document.querySelectorAll(".project");

  workBtnContainer.addEventListener("click", (e) => {
    projectContainer.classList.add("anim-out");

    const filter =
      e.target.dataset.filter || e.target.parentNode.dataset.filter;

    if (filter == null) {
      return;
    }

    addSelected(e.target);

    projects.forEach((project) => {
      if (filter === "all" || filter === project.dataset.type) {
        project.classList.remove("invisible");
      } else {
        project.classList.add("invisible");
      }
    });

    setTimeout(() => {
      projectContainer.classList.remove("anim-out");
    }, 1000);
  });
}
```

# function

## 4. Sending Email

Emailjs를 이용하여 나에게 email을 보낼 수 있다.

— CONTACT ME

Get In Touch

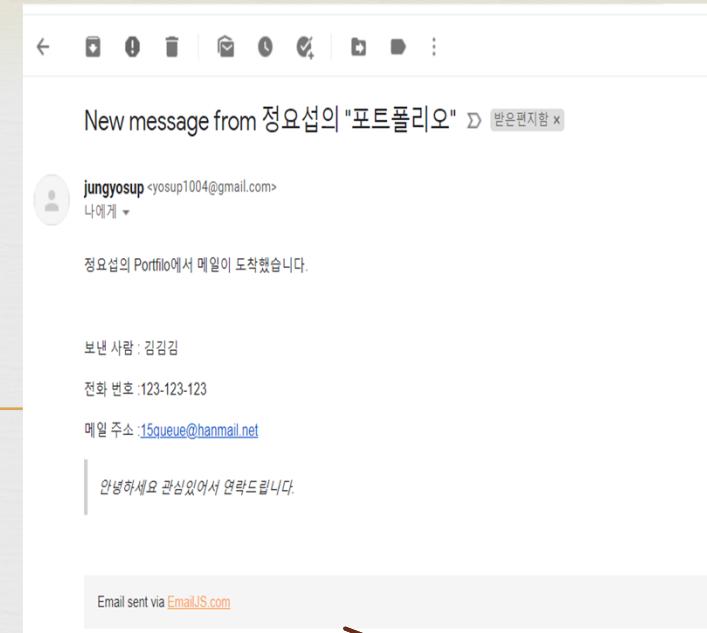
Name

Email

Phone number

Your Message

SEND MESSAGE



1. Emailjs를 이용하여  
나에게 email을 보낼 수  
있다.

```
// 13. get Email
const sendEmail = () => {
  (function () {
    emailjs.init("user_ESLwYmWgJ2he4Otypw2eN");
  })();

  const sendForm = document.querySelector("#contactbox_form");
  sendForm.addEventListener("submit", function (event) {
    event.preventDefault();
    emailjs.sendForm("ser", "tlatzi", "#contactbox_form").then(
      function (response) {
        console.log("SUCCESS!", response.status, response.text);

        if (response.status == 200) {
          successEmail();
        }
      },
      function (err) {
        console.log("FAILED...", err);
        console.log(err);
      }
    );
  });
};
```

```
// 14. success email

const successEmail = () => {
  const items = document.querySelectorAll(".contactbox_form_item");

  items.forEach((item) => {
    item.classList.add("success");

    setTimeout(() => {
      item.classList.remove("success");
    }, 1000);
  });
};
```

2. Email을 성공적으로  
보낼 시 성공여부를 CSS  
색상으로 판단할 수 있도  
록 하였다.

4

# Top Scorers

React 를 활용한  
각 리그 별 축구 득점 순위

URL

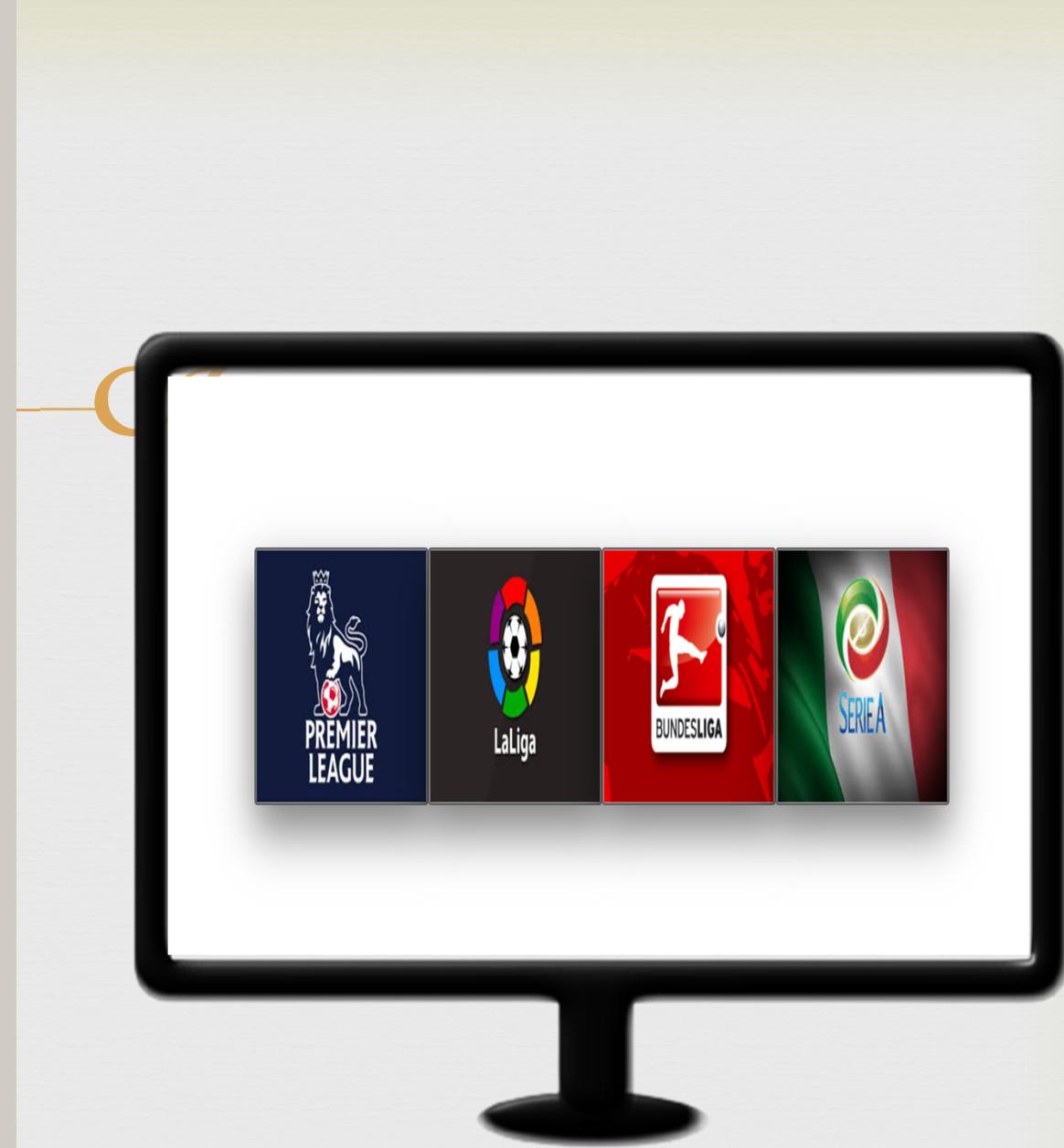
<https://youtu.be/yBgdU86bwnI>

특징

JavaScript 프레임워크 `react`를 활용  
`Axios`를 api 데이터 받기

구현언어

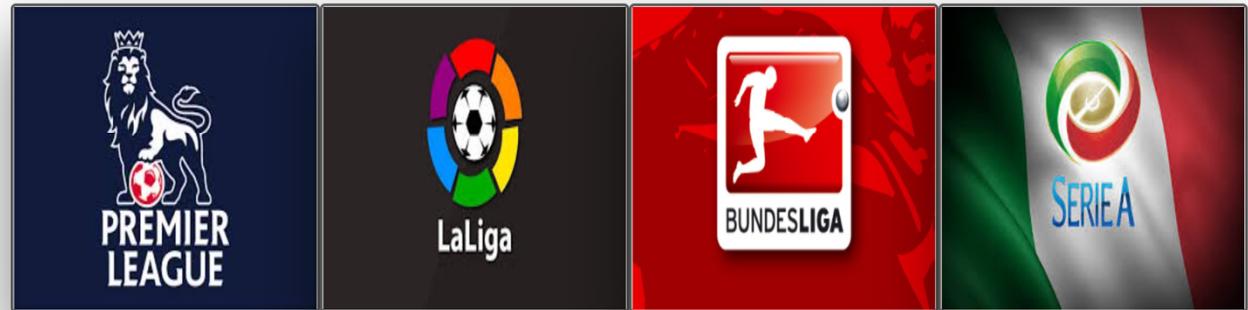
HTML, CSS, JavaScript



# Main Page

## HTML

BEM 방식을 사용



## Css

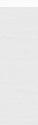
Flex를 활용한 Layout

## JavaScript



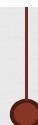
**Anchor**

영국 축구리그  
Top20명의 선  
수 명단 확인



**Anchor**

스페인 축구리그  
Top20명의 선  
수 명단 확인



**Anchor**

독일 축구리그  
Top20명의 선  
수 명단 확인



**Anchor**

이탈리아 축구리  
그 Top20명의  
선수 명단 확인

## Sub Page

### HTML

BEM 방식을 사용

### Css

Flex와 grid를 활용한 Layout

### JavaScript

React axios를 활용하여 Api  
를 받고, 받아온 데이터로 골을  
많이 넣은 순서대로 화면에 출력  
함

### Header

각 선수의 이름을 나타낸다.

### Nav

각 선수의 이미지를 보여준다.



D. Calvert-Lewin

**Born/age** 1997-03-16,23  
**Nationality** England  
**Height** 187 cm  
**Weight** 71 kg  
**Current club** Everton  
**goal** 10  
**assists**

[Detail Player](#)

Son Heung-Min

**Born/age** 1992-07-08,28  
**Nationality** Korea Republic  
**Height** 183 cm  
**Weight** 78 kg  
**Current club** Tottenham  
**goal** 9  
**assists** 2

[Detail Player](#)

Mohamed Salah

**Born/age** 1992-06-15,28  
**Nationality** Egypt  
**Height** 175 cm  
**Weight** 71 kg  
**Current club** Liverpool  
**goal** 8  
**assists** 1

[Detail Player](#)

J. Vardy

**Born/age** 1987-01-11,33  
**Nationality** England  
**Height** 179 cm  
**Weight** 74 kg  
**Current club** Leicester  
**goal** 8  
**assists** 2

[Detail Player](#)

### Section

### Content

각 선수들의 정보를 알려준다.

### Footer

React Link를 활용하여  
Section을 클릭시 상세페이지  
로 이동할 수 있다.

## Sub Page

### HTML

BEM 방식을 사용

### Css

Flex를 활용한 Layout

### JavaScript

React axios 를 활용하여 Api  
를 받고, 받아온 상세데이터로  
선수들의 상세정보를 볼 수 있게  
했다.

#### Player Data

Date of birth

1997-03-16

Place of birth

England Sheffield

Age

23

team

Everton



league

Premier League England



Position

Attacker

goals

10

assists

passes

197

rating

7.630000

#### detail

Axios로 받아온 선수들의 상세  
데이터를 활용하여 화면을 구성  
하였다.

# function

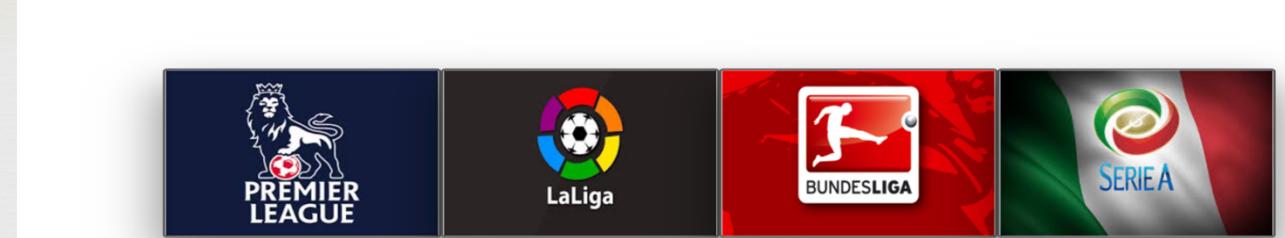
## 1. React router

React router를 활용하여 여러 페이지들이 존재하는 서비스를 만듦

```
import React from "react";
import { Link } from "react-router-dom";
import pre from "../assets/img/pre.jpg";
import bun from "../assets/img/bun.jpg";
import la from "../assets/img/la.jpg";
import seria from "../assets/img/seria.jpg";

function Navigation() {
  return (
    <div className="navbar">
      <Link to="/premier">
        <button className="navbar_btn">
          <img src={pre} alt="premier" className="navbar_img"/>
        </button>
      </Link>
      <Link to="/laliga">
        <button className="navbar_btn">
          <img src={la} alt="laliga" className="navbar_img"/>
        </button>
      </Link>
      <Link to="/bundesliga">
        <button className="navbar_btn">
          <img src={bun} alt="bundesliga" className="navbar_img"/>
        </button>
      </Link>
      <Link to="/seria">
        <button className="navbar_btn">
          <img src={seria} alt="seria A" className="navbar_img"/>
        </button>
      </Link>
    </div>
  );
}

export default Navigation;
```



1. React-router-dom  
의 Link를 활용하여 폐  
이지들이 존재하는 서비  
스를 구현

Player Data	
Date of birth	1997-03-16
Place of birth	England Sheffield
Age	23
team	Everton
league	Premier League England
Position	Attacker
goals	10
assists	
passes	197
rating	7.630000

```
return (
  <Link
    to={({ pathname: `/${league}/${id}`,
    state: { statistics, age, birth, } })}>
```

# function

## 2. Axios를 활용한 api 가져오기

Axios를 활용하여  
api-football-beta 데이터  
를 가져옴

```
class Premier extends React.Component {
  state = {
    isLoading: true,
    players: [],
    statistics: []
  };

  // League : 140 == lg
  // League : 39 == en
  getApi = async () => {
    const options = {
      method: "GET",
      url: "https://api-football-beta.p.rapidapi.com/players/topscorers",
      params: { season: "2020", league: "39" },
      headers: {
        "x-rapidapi-key": "fabb00016msh1ff7b0bb43cf770p146817jnsnc234adc2f0b9",
        "x-rapidapi-host": "api-football-beta.p.rapidapi.com",
      }
    };

    // 플레이어의 정보를 가져온다.
    const players = await axios
      .request(options)
      .then(function (response) {
        console.log(response.data);
        const array = response.data.response.map((data) => data.player);
        return array;
      })
      .catch(function (error) {
        console.error(error);
      });

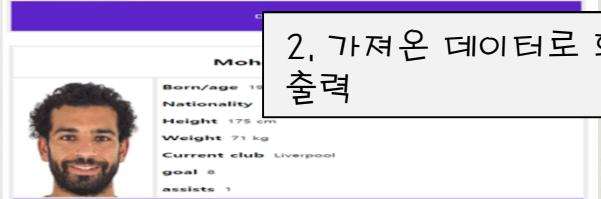
    // 플레이어의 상세정보를 가져온다.
    const statistics = await axios
      .request(options)
      .then(function (response) {
        console.log(response.data);
        const filter = response.data.response.map((data) => data.statistics);
        const array = filter.map((data) => {
          let count = 0;
          return data[count++];
        });
        console.log(array);
        return array;
      })
      .catch(function (error) {
        console.error(error);
      });

    this.setState({
      isLoading: false,
      players,
      statistics,
    });
  };
}
```

```
(20) [{}]
  0: {id: 186, name: "D. Calvert-Lewin", firstname: "Dominic", lastname: "Calvert-Lewin", age: 23, ...}
  1: {id: 186, name: "Son Heung-Min", firstname: "Heung-Min", lastname: "Son", age: 28, ...}
  2: {id: 19134, name: "P. Bamford", firstname: "Patrick", lastname: "Bamford", age: 27, ...}
  3: {id: 306, name: "Mohamed Salah", firstname: "Mohamed", lastname: "Salah Ghaly", age: 28, ...}
  4: {id: 18786, name: "J. Vardy", firstname: "Jamie", lastname: "Vardy", age: 33, ...}
  5: {id: 184, name: "H. Kane", firstname: "Harry", lastname: "Kane", age: 27, ...}
  6: {id: 1405, name: "Bruno Fernandes", firstname: "Bruno", lastname: "Fernandes", age: 26, ...}
  7: {id: 2939, name: "C. Wilson", firstname: "Callum", lastname: "Wilson", age: 28, ...}
  8: {id: 19366, name: "O. Watkins", firstname: "Ollie", lastname: "Watkins", age: 25, ...}
  9: {id: 19187, name: "J. Grealish", firstname: "Jack", lastname: "Grealish", age: 25, ...}
  10: {id: 2678, name: "Diogo Jota", firstname: "Diogo", lastname: "Teixeira da Silva", age: 24, ...}
  11: {id: 18955, name: "Daniel William John Ings", firstname: "Daniel William John", lastname: "Ings", age: 28, ...}
  12: {id: 3247, name: "W. Zaha", firstname: "Wilfried", lastname: "Zaha", age: 28, ...}
  13: {id: 1166, name: "T. Werner", firstname: "Tim", lastname: "Werner", age: 24, ...}
  14: {id: 2398, name: "J. Ward-Prowse", firstname: "James", lastname: "Ward-Prowse", age: 26, ...}
  15: {id: 19428, name: "J. Bowen", firstname: "Jarrod", lastname: "Bowen", age: 24, ...}
  16: {id: 384, name: "S. Hand", firstname: "Sadio", lastname: "Hand", age: 28, ...}
  17: {id: 635, name: "R. Mahrez", firstname: "Riyad", lastname: "Mahrez", age: 29, ...}
  18: {id: 2887, name: "R. Jiménez", firstname: "Raúl Alonso", lastname: "Jiménez", age: 29, ...}
  19: {id: 2726, name: "K. Zouma", firstname: "Kurt Happy", lastname: "Zouma", age: 26, ...}
  length: 20
  >__proto__: Array(0)
```

```
(20) [{}]
  0: {team: {}, league: {}, games: {}, substitutes: {}, shots: {}}
  1: {team: {}, league: {}, games: {}, substitutes: {}, shots: {}}
  2: {team: {}, league: {}, games: {}, substitutes: {}, shots: {}}
  3: {team: {}, league: {}, games: {}, substitutes: {}, shots: {}}
  4: {team: {}, league: {}, games: {}, substitutes: {}, shots: {}}
  5: {team: {}, league: {}, games: {}, substitutes: {}, shots: {}}
  6: {team: {}, league: {}, games: {}, substitutes: {}, shots: {}}
  7: {team: {}, league: {}, games: {}, substitutes: {}, shots: {}}
  8: {team: {}, league: {}, games: {}, substitutes: {}, shots: {}}
  9: {team: {}, league: {}, games: {}, substitutes: {}, shots: {}}
  10: {team: {}, league: {}, games: {}, substitutes: {}, shots: {}}
  11: {team: {}, league: {}, games: {}, substitutes: {}, shots: {}}
  12: {team: {}, league: {}, games: {}, substitutes: {}, shots: {}}
  length: 12
  >__proto__: Array(0)
```

1. Axios를 활용하여  
api를 가져왔다.



2. 가져온 데이터로 화면  
출력

# function

## 3. React Spinner

생명주기를 이용해 React spinner를 활용하여 데이터를 다 받아올 때까지 이미지로 로딩 표시

```
class Premier extends React.Component {
  state = {...};

  // League : 140 == La...
  getApi = async () => { ... };

  componentDidMount() { ... }

  render() {
    const { isLoading, players, statistics } = this.state;
    console.log(isLoading);
    console.log(players);
    console.log(statistics);
    let count = 0;

    return (
      <section className="container">
        {isLoading ? (
          <div className="loader">
            <Spinner />
          </div>
        ) : (
          <div>
            <div className="players">
              <div className="topscorers">
                <h1>2020년도 Premier Best 20 Players</h1>
              </div>
              {players.map((player) => (
                <Player
                  key={player.id}
                  id={player.id}
                  age={player.age}
                  birth={player.birth}
                  height={player.height}
                  name={player.name}
                  nationality={player.nationality}
                  photo={player.photo}
                  weight={player.weight}
                  statistics={statistics[count++]}
                  league={"premier"}
                />
              )));
            </div>
          </div>
        )}
      </section>
    );
  }
}
```

3. State에 받아온 데이터가 들어감

2. componentDidMount()가 작동하고 Api를 가져오고 setState로 state에 데이터를 넣음

1. render()가 동작, isLoading이 true이므로 Spinner Api가 작동

4. 다시 render()가 동작하면서 받아온 데이터로 화면에 출력

4

# 면접학교

## 웹사이트 제작

URL

<https://youtu.be/Kd3G7OvXQmQ>

특징

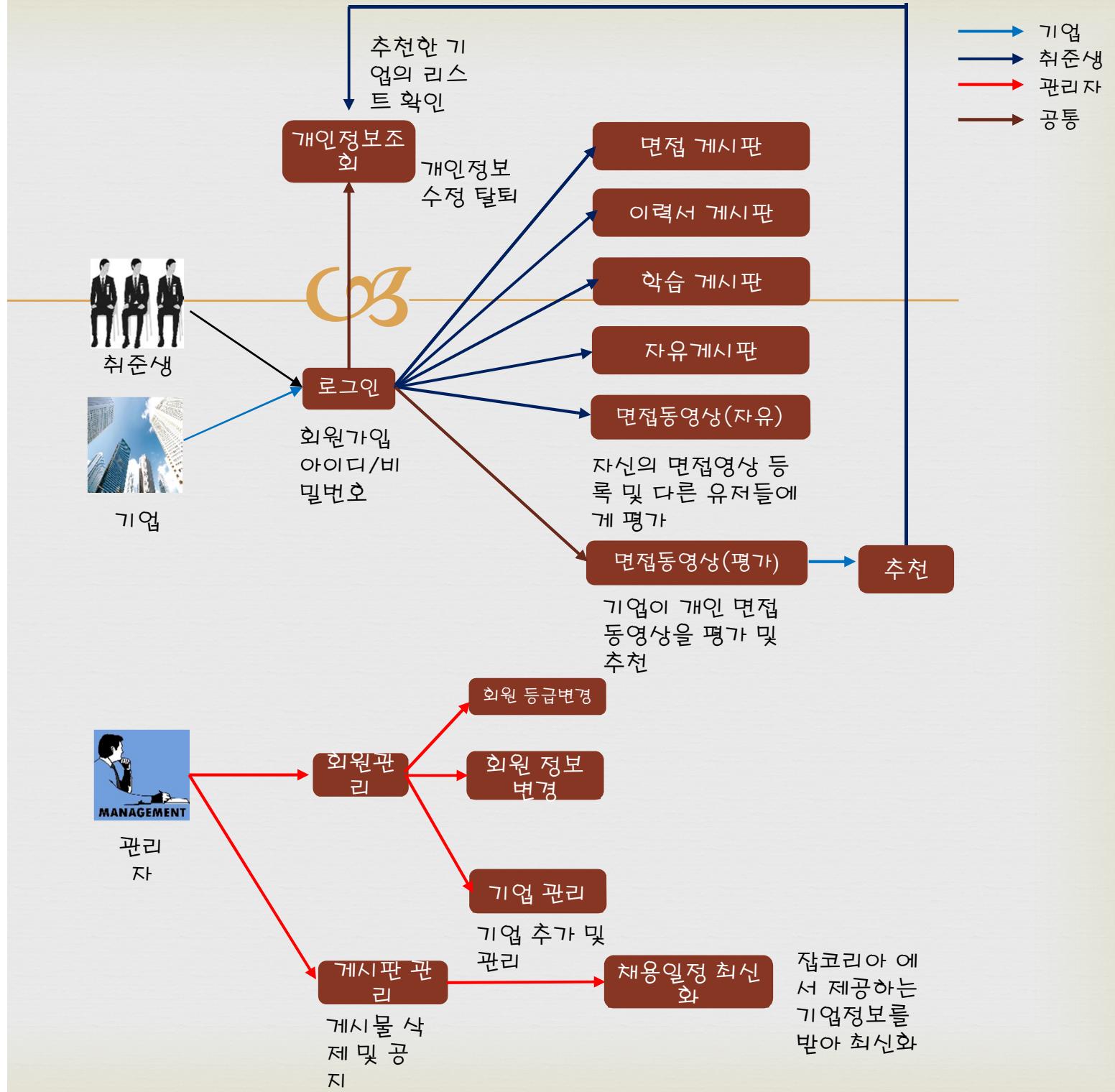
"취준생을 위한" 웹 사이트

구현언어

HTML, CSS, JavaScript, Java



# Navigation flow



## Main Page

HTML

Css

JavaScript

JQuery 사용

Java

HTML 코드에 Java코드를 넣어  
동적인 웹 페이지를 생성(JSP)



## Header

메인 페이지로 이동하는 링크  
버튼

검색기능

Navbar 클릭 시 Down Drop  
이 나옴

## Main

각 창업 및 면접 관련된 정보가  
자동 슬라이더로 중간하단의 컨  
트롤 버튼으로 제어

Navbar 클릭 시 Dropdown이  
나옴

각 게시판에 게시글을 적을 시  
Main Page에 5개씩 나오게끔  
구현

기존 로그인 및 회원가입 추가  
적으로  
카카오 톡으로 자동 회원가입  
할 수 있도록 연동

## Sub Page

### HTML

### Css

### JavaScript

### JQuery 사용

### Java

HTML 코드에 Java 코드를 넣어  
동적인 웹 페이지를 생성(JSP)

면접학교

일반회원 기업회원

아이디	<input type="text"/>	중복체크
비밀번호	<input type="password"/>	
비밀번호확인	<input type="password"/>	
이름	<input type="text"/>	
주소	<input type="text"/> 우편번호 찾기 <input type="text"/> 우편번호	<input type="text"/> 주소 <input type="text"/> 상세주소
생년월일	연도-월-일 <input type="button" value=""/>	
전화번호	<input type="text"/>	
이메일주소	<input type="text"/>	인증하기
인증번호 : <input type="text"/>		
성별	<input type="radio"/> 여자 <input type="radio"/> 남자	
직종	<input type="text" value="회사원"/>	

가입완료 취소

## Header

일반회원과 기업회원 분리된 회  
원가입

## Main

일반적인 회원가입 창

중복체크 및 이메일 주소 인증  
하는 기능 구현

## Sub Page

HTML

Css

JavaScript

JQuery 사용, bootstrap 사용

Java

HTML 코드에 Java코드를 넣어  
동적인 웹 페이지를 생성(JSP)

The screenshot shows a web page titled '면접학교' (Interview School). At the top, there is a search bar with dropdown menus for '전체검색' (Search All) and '검색어를 넣어주세요' (Please enter your search term), and a '검색' (Search) button. Below the search bar is a navigation menu with links to '소개' (About), '동영상' (Video), '면접게시판' (Interview Board), '이력서게시판' (CV Board), '학습게시판' (Learning Board), '자유게시판' (Free Board), and 'Q&A'. The main content area displays a table of video posts under the heading '유저동영상' (User Videos) and '기업동영상' (Corporate Videos). The table has columns for '번호' (Number), '제목' (Title), '작성자' (Author), '작성일' (Date), '조회' (Views), and '좋아요' (Likes). Two posts are listed:

번호	제목	작성자	작성일	조회	좋아요
70	안녕하세요 정요섭입니다.[0]	정요섭	2018-12-31 18:10:08	1	0
64	안녕하세요 김철수 면접영상입니다.[0]	정요섭	2018-12-31 17:58:39	0	0

Below the table is a '글쓰기' (Write a post) button and a dropdown menu for '제작내용' (Content Type) with options like '동영상' (Video), '문서' (Document), and '링크' (Link). A '검색' (Search) button is also present.

This screenshot shows the '동영상게시판' (Video Board) interface. It features a toolbar at the top with various icons for file operations. Below the toolbar is a text input field for '제작분야' (Category) and a dropdown menu for '제작' (Production). The main area contains a rich text editor with a WYSIWYG interface, including a toolbar with icons for bold, italic, underline, etc., and a preview window showing the current content. At the bottom, there are buttons for '등록' (Register) and '취소' (Cancel).

This screenshot shows a video player embedded in a web page. The video title is '안녕하세요 정요섭입니다.' and the subtitle is '20181231181002\_정요섭\_자기소개.mp4'. Below the video, there is a summary of the post: '제작' (Production) is '정요섭', '작성일' (Date) is '2018-12-31 18:10:08', and '조회수' (Views) is '1'. There are also buttons for '답글' (Reply) and '목록' (List).

## Header

메인 페이지로 이동하는 링크  
버튼, 검색기능

## Main

자신의 면접 영상을 올리는 게  
시판으로 면접영상을 보고 면접  
관들의 조언을 얻음

## Sub Page

HTML

CSS

JavaScript

JQuery 사용, bootstrap

Java

HTML 코드에 Java 코드를 넣어  
동적인 웹 페이지를 생성(JSP)

The screenshot shows a web page titled '면접학교' (Interview School). At the top, there are tabs for 소개 (Introduction), 동영상 (Video), 면접게시판 (Interview Bulletin Board), 이력서게시판 (Curriculum Vitae Bulletin Board), 학습게시판 (Learning Bulletin Board), 자유게시판 (Free Bulletin Board), and Q&A. Below the tabs is a search bar with a dropdown menu for '전체검색' (Search All) and a button to '검색' (Search).  
  
The main content area features a calendar for November 2018. The days of the week are labeled from 일 (Monday) to 토 (Sunday). The weeks are numbered 44 through 49. Each day cell contains a company name and its interview date. Some cells are highlighted in blue, indicating specific events or sections.  
  
Key highlights from the calendar:

- Week 44: Friday, November 30, (주)리큅트루 (주)나임세인씨 (주)글로벌네트웍스
- Week 45: Saturday, December 1, (주)미디어월드네트웍스 (주)미디어카리어 (주)에스오월드테크놀 (주)한국지식중유협회 주 (주)잡뉴스솔루션사 (주)에나름 (주)디스파 (주)카스랑프 (주)스카우트풀센  
+5개
- Week 46: Sunday, December 2, (주)제이콥시스템 (주)모아저축은행 (주)아큐라소프트 (주)케이원시스템  
+9개
- Week 47: Saturday, December 8, (주)모모디아카데미 (주)모모디아카데미 (주)모모디아카데미 (주)모모디아카데미  
+12개
- Week 48: Sunday, December 16, (주)로딕스 (주)비즈풀 (주)트래블로지 (주)스마트프로 (주)로딕스 (주)비즈풀 (주)트래블로지 (주)스마트프로 (주)로딕스 (주)비즈풀 (주)트래블로지 (주)스마트프로  
+87개
- Week 49: Saturday, December 22, (주)클로우도 (주)리티스글로벌커뮤 (주)로리아센터 (주)아이디어很差 (주)로리아센터 (주)로리아센터

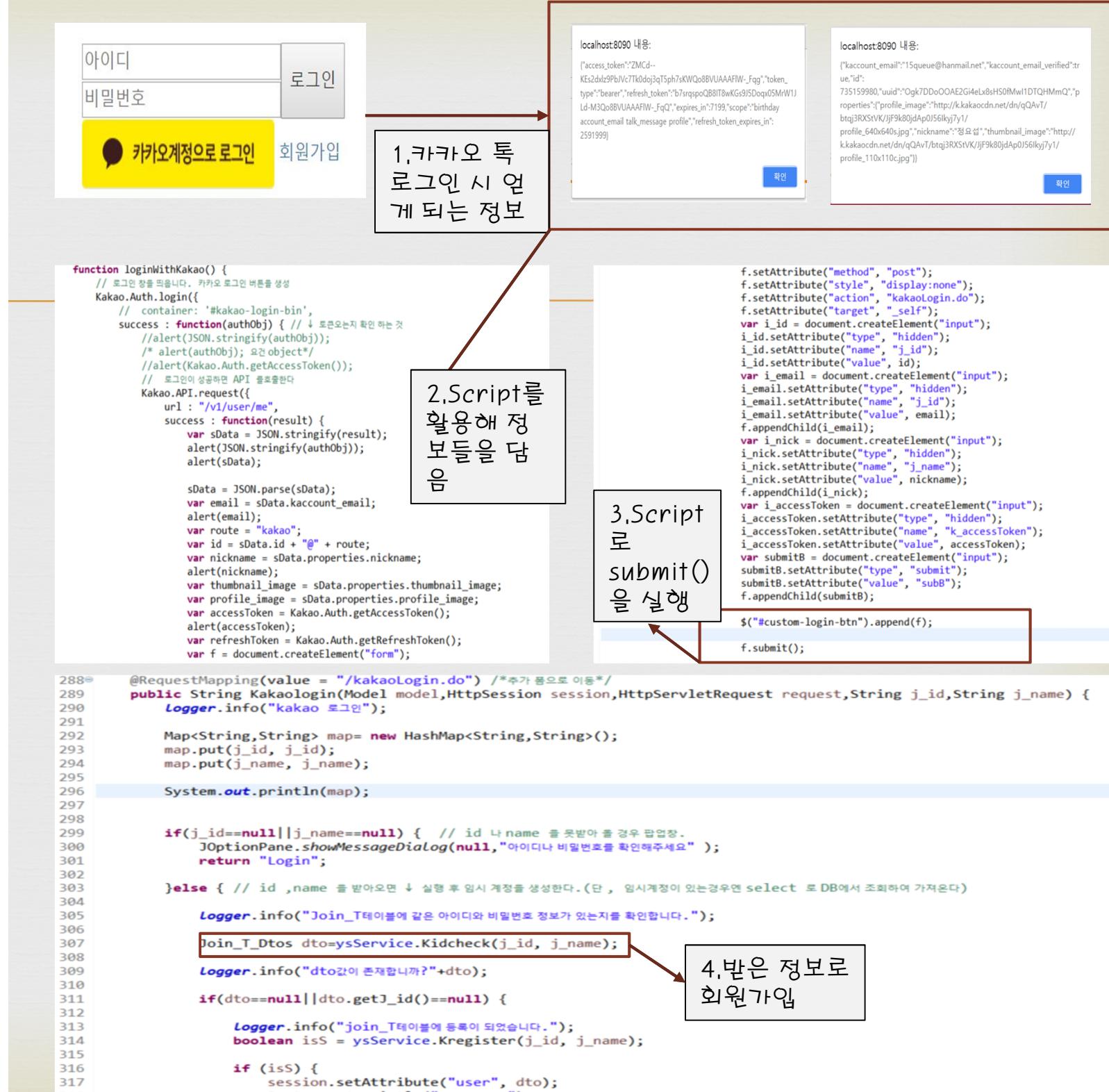
## Main

각 기업의 공채일정을 확인할  
수 있는 달력

# function

## 1. KaKaoTalk Login

KaKaoTalk Api를 활용하여  
카카오 톡으로도 로그인 가능  
하게 함



# function

## 2. 아이디 중복체크

회원가입을 위해 임의의 값을 e-mail로 보내 e-mail 인증을 하게끔 함

아이디	<input type="text"/> 중복체크
비밀번호	<input type="password"/>
비밀번호확인	<input type="password"/>
이름	<input type="text"/>
주소	<input type="text"/> 우편번호 찾기 <input type="text"/> 우편번호 <input type="text"/> 주소 <input type="text"/> 상세주소
생년월일	<input type="text"/> 연도.월.일
전화번호	<input type="text"/>
이메일주소	<input type="text"/> 인증하기 <input type="text"/> 인증번호:
성별	여자 <input checked="" type="radio"/> 남자 <input type="radio"/>
직종	회사원 <input type="button" value="▼"/>

4. 아이디 존재여부에 따라 리턴 받은 값을 if문과 JavaScript를 통해 중복여부 값을 넣어줌

```
//아이디중복확인
function chkDupId() {
    var prmId = $('#insertId').val();
    if(prmId==''){
        $('#idid').html('ID를 입력해주세요!!');
    }else if(!(prmId>='0'&&prmId<='9'||prmId>='a'&&prmId<='z')){
        $('#idid').html('숫자와 영어소문자만 사용해주세요.');
    }else{
        $.ajax({
            type:'POST',
            data:"j_id="+prmId,
            dataType:'text',
            url:'chkDupId.do',
            success:function(rData,textStatus,xhr){
                var chkRst=rData;
                if(chkRst==0){
                    $('#idid').html('가능한 ID 입니다.');
                    $('#idChk').val('Y');
                }else{
                    $('#idid').html('중복된 ID 입니다.');
                    $('#idChk').val('N');
                }
            },
            error:function(xhr,status,e){
                alert(e);
            }
        });
        return;
    }
}
```

1. 데이터 베이스에 중복되는 아이디가 있는지 체크하기 위해 controller로 입력한 아이디를 보냄

```
@RequestMapping(value = "/chkDupId.do", method = RequestMethod.POST)
public void checkId(HttpServletRequest req, HttpServletResponse res, ModelMap model) throws Exception {
    PrintWriter out = res.getWriter();
    System.out.println(out);
    try {
        //넘어온 ID를 받는 곳
        String paramId=(req.getParameter("j_id")==null)?String.valueOf(req.getParameter("j_id"));
        dto.setId(paramId.trim());
        System.out.println(dto.getId());
        int chkPoint=Service_Regist.chkDupId(dto);
        System.out.println(chkPoint);
        System.out.println(dto);
        out.print(chkPoint);
        out.flush();
        out.close();
    } catch (Exception e) {
        e.printStackTrace();
        out.print("1");
    }
}
```

2. 받은 아이디를 데이터베이스에 보내줌

```
<select id="chkDupId" parameterType="Join_T_Dtos" resultType="Integer">
    SELECT COUNT(*) AS J_ID FROM JOIN_T WHERE J_ID=#{j_id}
</select>
```

3. 데이터베이스에서 쿼리 문을 통해 아이디가 존재여부를 판단

# function

## 3. E-mail 인증

회원가입을 위해 임의의 값을 e-mail로 보내 e-mail 인증을 하게끔 함

아이디	<input type="text"/>	<input type="button" value="중복체크"/>
비밀번호	<input type="password"/>	
비밀번호확인	<input type="password"/>	
이름	<input type="text"/>	
주소	<input type="button" value="우편번호 찾기"/>	<input type="button" value="우편번호"/>
	<input type="text" value="주소"/>	<input type="text" value="상세주소"/>
생년월일	<input type="text" value="연도-월-일"/>	
전화번호	<input type="text"/>	
이메일주소	<input type="text"/>	<input type="button" value="인증하기"/>
	<input type="text" value="인증번호"/>	
성별	여자 <input checked="" type="radio"/> 남자 <input type="radio"/>	
직종	<input type="text" value="회사원"/>	

```
<!-- 이메일인증 api -->
<script src="https://smtpjs.com/smtp.js"></script>
<script type="text/javascript">
$(function () {
    $("#auth_btn").click(function () {
        var data = {"email": $("#email").val()};
        var authNum = "";
        //입력한 이메일 값은 불러온 동일색
        var emailsIn = document.getElementById('email').value;
        $.ajax({
            url : "emailAuth.do",
            data : data,
            type : "POST",
            success : function (data) {
                authNum = data;
                //이메일을 보내는 소스
                Email.send("yoyoyosup@naver.com",
                           emailsIn,
                           "면접커뮤니티 가입인증 확인 메일입니다.",
                           "인증번호를 확인 후 인증 화면에 입력해 주세요."+" "+"인증번호"+ "=" : "+authNum,
                           "smtp.naver.com",
                           "yoyoyosup");
            }
        });
    });
}); // 이메일 인증 버튼 end
```

1. 컨트롤러로 내가 입력한 이메일 주소를 보내준다.

```
//인증번호
@RequestMapping(value="/emailAuth.do" , produces="text/plain;charset=utf-8",method = RequestMethod.POST)
@org.springframework.web.bind.annotation.ResponseBody
public String emailAuth(javax.servlet.http.HttpServletRequest request) {
    String email = request.getParameter("email");
    String authNum = "";
    System.out.println(email);
    authNum = randomNum();
    //가입승인에 사용될 인증키 난수 발생
    String str = authNum;
    System.out.println(str);
    return str;
}
```

2. 이메일 주소를 받은 컨트롤러는 가입승인에 사용될 인증 키를 return해준다.

3. 리턴 받은 인증 키를 입력한 이메일주소로 보내고 받은 인증 키와 내가 입력하고 인증번호가 같을 시 인증완료

# function

#### 4. 다음에디터게시판

다음 에디터 API를 활용하여  
게시판 구현 및 댓글, 좋아요,  
즐겨찾기 기능 구현

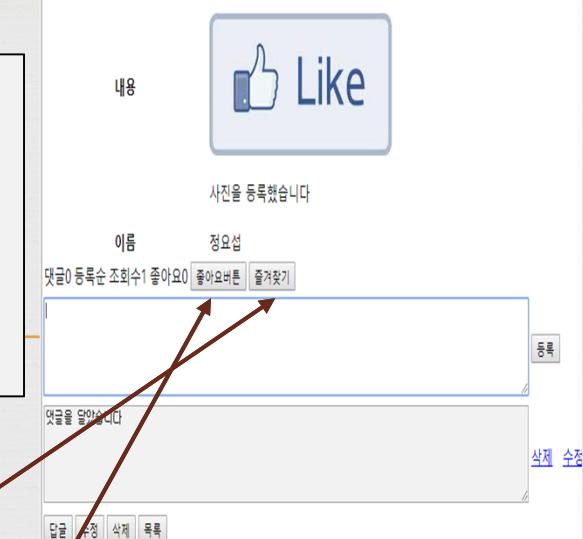
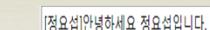
```
/* 넘겨 받아야 할 값들이 게시판이름, 회원번호, 게시판번호, 글내용, */  
function reply(seq.unum.boardname) {
```

```
//14. 즐겨찾기를 가져오는 메서드
@Override
public Funtion_T_Dtos selectsearch(Answer_T_Dtos dto) {
    return SqlSession.selectOne(namespace+"selectsearch",dto);
}

//15. 즐겨찾기를 삭제하는 메서드
@Override
public boolean deletesearch(Answer_T_Dtos dto) {
    int count =0;
    count = SqlSession.delete(namespace+"deletesearch", dto);
    return count>0?true:false;
}
```

```
function like(seq,unum,f_like) {
$.ajax({
    url: "likeboard.do",
    data: "a_seq="+seq+"&a_unum="+unum,
    type: "post",
    datatype: "json",
    success: function(name(result) {
        var result = result["f_saveorlike"];
        if(result !=null){
            $("#likeup").html("좋아요"+result.f_like);
        }
        if(result==null&&f_like=='0'){
            $("#likeup").html("좋아요"+(f_like));
        }else if(result==null){
            $("#likeup").html("좋아요"+(f_like));
        }
    }
});
```

댓글을 달기  
위해 script  
를 사용하여  
화면에 댓글  
창이 나타나  
게 한다



정요섭님(일반회원)환영

내정보 | 로그아웃

즐겨찾기한 게시판

- 게시글:3 글제목:아녕하세요 정요섭입니다.detailboard.do

좋아요 기능은 한번만 누르도록 제한을 준다

좋아요, 즐겨 찾기를  
누를 시 DB로 그 누  
른 여부가 전달되고  
내정보에서 확인할  
수 있다.

# function

## 5. 채용일정 확인

잡코리아에서 제공하는 기업 정보를 받아 DB에 전달하고 그 값을 달력에 나타나게 함

```

<script type="text/javascript">
$(function(){
    $("button").click(function(){
        $.ajax({
            url:"jobAjax.do",
            data:"keywords=api.saramin.co.kr/job-search?keywords=%EC%9B%99%EA%B0%9C%EB%80%9C%EC%9E%98&fields=posting-date+expiration-date&sort=ud&start=1",
            type:"POST",
            datatype:"json",
            success:function(doc){
                for (var i = 0; i < 110; i++) {
                    var tr1=document.createElement("tr"); //<tr>/tr>

```

1. Ajax를 활용하여 잡코리아의 기업 정보를 받고 그 값을 JavaScript를 활용하여 HTML에 뿌려준다.

```

<body>
<form action="cominfo.do" method="POST">
    <table border="1" id="cop">
        <tr>
            <td>회사명</td>
            <td>공고명</td>
            <td>등록일</td>
            <td>채용마감일</td>
        </tr>
    </table>
    <input type="submit" value="정보전송"><!-- C_List -->
</form>
<button id="saram1">채용정보확인</button>
</body>

```

2. 그 정보를 form 태그를 이용해 데이터 베이스에 저장

CO_SEQ	CO_NAME	CO_TITLE	CO_LOCATION	CO_JOBTYPE
1	(주)알리스	취업준비/창업최적/풀기획/웹디자인/UX/서울 > 마포구	정규직, 인턴직,프리랜서,인턴직 (경)	
2	(주)알리스	기획자/디자이너/마케터/취업준비/창업최적 서울 > 마포구	정규직, 인턴직,프리랜서,인턴직 (경)	
3	(주)케이맨파워	[최고대우] BC 웹기획 (3년이상)	정규직	
4	(주)리아시안	[보내인증 기술지원] 솔루션 기술 지원 및 연 서울 > 강남구	정규직	
5	(주)리엔소프트코리아/에이티비	디자일레이션/리엔소프트에서 PHP, JAVA, 서울 > 강남구	정규직	
6	주식회사 멘리프트데케이션	PHP 정규직&계약직 재용(경력/신입)	정규직, 계약직,기간제	
7	와이즈아프엔	Android, IOS 플랫폼 개발자 모집	정규직	
8	(주)크리.Tools	경력 프론트엔드 개발자 모집	정규직	
9	(주)크리.Tools	글꼴 IT 업무 기획 및 분석 경력자 채용	정규직	
10	(주)알리드아이비이에스	[자기소개] 위드아노메이션 개발자군 채용	정규직	
11	주식회사 제이비스웨어	콘텐츠 마케팅 그룹 2018 상반기 백엔드 개발대구 > 동구	계약직,계약직 (정규직 전환가능)!	
12	(주)유먼디자일	상장사 SIP 프로토콜 혹은 VoIP 업무 - STL, 서울 > 영등포구	정규직	
13	엔테크서비스 주식회사	웹서비스 개발 경력직 채용	서울 > 금천구 경기 > 성남시 분당구 정규직	
14	(주)시나리오	웹/모바일 시나리오 개발자/UI/UX/UI 저그치 서초 > 서초구	저그치	

```

//클릭으로 이동하는 메서드
@RequestMapping(value = "/calendarform.do")
public String Calendarform(Locale locale, Model model, String a_boardname) {
    System.out.println("여기로 이동을 안하나요?");
    List<cominfo_Dto> comName = cominfo_service.getComInfo();
    model.addAttribute("comName", comName);
    model.addAttribute("a_boardname", a_boardname);
    return "Calendarform";
}

```

3. 데이터 베이스에 저장된 값을 controller를 이용해 달력에 뿌려줌

