

## I. HTML

### 1) HTML이란?

HTML은 Hyper Text Markup Language 약어로 HyperText(웹 페이지에서 다른 페이지로 이동할 수 있도록 하는 것) 기능을 가진 문서를 만드는 언어입니다. 다시 말해, 구조를 설계할 때 사용되는 언어로 hyper link 시스템을 가지고 있으며, 흔히 말하는 웹 페이지를 위한 마크업 언어라고 할 수 있습니다.

HTML은 웹페이지의 구조를 결정합니다. 이 구조만으로는 웹페이지를 보기 좋게 꾸미거나 상호작용이 가능하게 만들기는 어렵습니다. 그렇기 때문에 HTML을 예쁘게 꾸미거나 상호작용이 가능하게 하기 위해서는 각각 CSS와 자바스크립트라는 보조 기술을 사용하게 됩니다.

### 2) HTML 태그

HTML 문서의 모양과 행동양식을 정해주는 명령어 이름으로 <> 속에 HTML 태그 명령어의 이름을 작성하는 형태로 사용합니다. 따라서 자신이 사용하고자 하는 기능을 가진 HTML 태그 명령어를 <> 속에 작성하여 HTML 문서를 작성할 수 있습니다. 예를 들어 한 줄을 띄우는 HTML 태그 명령어는 br이라고 하는데, 실제로 사용할 때 <br>의 형태로 작성해야 합니다.

### 3) HTML 요소

한 HTML 요소는 시작 태그(opening tag), 글자, 내용, 종료 태그(closing tag)로 이루어져 있습니다. HTML 요소의 내용이 비어 있을 때도 있습니다. 종료 태그 대신 웹페이지에 삽입하려는 콘텐츠의 소스나 링크가 있는 경우입니다. 비어 있는 HTML 요소의 예로는 웹페이지에 이미지를 삽입할 때 쓰는 <img>가 있습니다. HTML 요소는 종종 태그와 호환해서 사용되지만, 둘 사이에는 약간의 차이점이 있습니다. HTML 요소는 한 쌍의 시작 태그와 종료 태그, 그리고 그 사이에 콘텐츠가 포함된 조합을 말합니다.

### 4) HTML 속성

HTML 태그에도 속성이라 불리는 것이 있습니다. 이 속성들은 시작 태그 안에 위치하며, 스타일 및 id부터 class까지 다룹니다. 이들은 요소에 관한 추가 정보를 전달하고 스타일 지정 및 자바스크립트를 사용한 조작과 같은 작업의 수행을 돕는 속성값을 사용합니다.

### 5) 시맨틱 HTML이란?

시맨틱 단어 그 자체에는 '의미의, 의미론적인' 라는 뜻이 담겨있습니다. 이로 유추해보았을 때, 시맨틱 태그는 태그 내용에 의미를 부여하는 태그라고 할 수 있습니다. 시맨틱 태그는 웹 페이지에 보이는 것 이상의 정보를 제공합니다.

시맨틱 태그(Semantic Tag)는 포함된 콘텐츠의 특정 의미를 정의하고 목적을 갖는 태그입니다. 기존 HTML <div> 태그의 기능과 마찬가지로 block element이면서 사이트의 구조(레이아웃)을 설계하기 위해 존재합니다. 시맨틱 태그의 요소로는 <header>, <nav>, <article>, <section>, <footer>, <main> 등이 존재합니다. 이러한 시맨틱 태그 요소는 콘텐츠를 논리적 섹션으로 구성하고 각 부분의 역할과 기능을 전달하는 데 도움이 됩니다. 다시 말해, 시맨틱 태그는 HTML의 구조를 설계하는데 있어 태그에 의미를 부여함으로써 웹사이트의 구조를 파악하기 쉽도록 도와주기 위해 만들어진 것입니다.

## II. CSS

### 1) CSS란?

종속형 시트 또는 캐스케이딩 스타일 시트(영어: Cascading Style Sheet)는 마크업 언어가 실제 표시되는 방법을 기술하는 스타일 언어(영어: Style sheet language 스타일 시트 랭귀지[\*])로[1], HTML과 XHTML에 주로 쓰이며, XML에서도 사용할 수 있습니다. W3C의 표준이고, 레이아웃과 스타일을 정의할 때의 자유도가 높습니다. 기본 파일명[2]은 style.css입니다. 글꼴이나, 배경색, 너비와 높이, 위치 등을 지정하거나, 웹 브라우저, 스크린 크기, 장치에 따라서 화면을 다르게 표시될 수 있도록 지정할 수도 있습니다. 즉, HTML 구조는 그대로 두고 CSS 파일만 변경해도 전혀 다른 웹사이트처럼 꾸밀 수 있습니다.

### 2) CSS 문법

CSS는 단순한 문법을 가지며, 수많은 영어 키워드를 사용하여 다양한 스타일의 프로퍼티의 이름을 규정합니다. 스타일 시트는 규칙의 목록으로 구성된다. 각 규칙이나 규칙 집합은 하나 이상의 셀렉터와 하나의 선언 블록을 이룹니다.

#### - 선택자

CSS에서 선택자, 또는 셀렉터(영어: Selector)는 어느 부분의 마크업에 스타일을 적용할지 선언함

니다. 태그명(태그의 이름)과 클래스, ID로 선택할 수 있습니다. 클래스와 ID들은 대소문자를 구분하며 문자로 시작해야 하고 영숫자와 언더바(\_)를 포함할 수 있습니다. 클래스는 어떠한 요소의 어떠한 수의 인스턴스에도 적용할 수 있으며, ID는 하나의 요소에만 적용할 수 있습니다. ID 값은 앞에 #이 붙고, 클래스 값은 앞에 .이 붙습니다.

#### - 선언 블록

선언 블록은 괄호로 이루어진 선언들의 목록으로 이루어집니다. 각 선언은 그 자체가 프로퍼티, 쌍점(;), 값으로 구성됩니다. 한 블록 안에 여러 선언이 있으면, 쌍반점 (;, 세미콜론)이 개개의 각 선언에 삽입되어야 하며, [8] 단 하나의 선언이라도 세미콜론을 넣지 않으면 동작하지 않는 경우도 있습니다.

### 3) CSS로 기대되는 효과

첫째, 정보(HTML)와 디자인(CSS)을 분리하여 관리할 수 있습니다.

둘째, 정보를 수정하지 않고 디자인만 변경할 수 있습니다. 태그마다 style 속성으로 주게 되면 소스 코드가 지저분해 지는데, 분리하여 관리하게 되면 유지보수 및 가독성이 좋아집니다.

셋째, 검색엔진이 HTML을 해석 가능하도록 하기 때문에 더 많은 방문자들이 방문할 수 있도록 유도할 수 있습니다. 시각장애인을 위한 스크린리더 프로그램이 HTML문서를 해석하도록 할 수 있습니다. (웹접근성)

## III. js

### 1) js란?

JavaScript는 개발자가 대화식 웹 페이지를 만들기 위해 사용하는 프로그래밍 언어입니다. 자바스크립트는 1995년에 Netscape Navigator 2.0 이라는 브라우저에 탑재되어 나오므로써 웹 프로그래밍의 개념이 창시되었습니다. 현재는 모든 브라우저에서 실행된다. 자바스크립트는 HTML 페이지에 소스 코드가 조각난채로 내장되어있으며, 따로 컴파일 과정을 필요로 하지 않습니다. 각 브라우저 내부의 자바스크립트 처리기(interpreter)에 의해서 실행됩니다. 소셜 미디어 피드 새로고침부터 애니메이션 및 대화형 지도 표시에 이르기까지, JavaScript 기능은 웹사이트의 사용자 경험을 개선할 수 있습니다. 클라이언트 측 스크립팅 언어로서 JavaScript는 월드 와이드 웹의 핵심 기술 중 하나입니다. 예를 들어 인터넷 탐색 시 웹페이지에서 이미지 슬라이드쇼, 클릭하면 표시되는 드롭 다운 메뉴 또는 객체 색상의 동적 변화를 보게 된다면 이는 JavaScript의 효과를 보는 것입니다.

### 2) JavaScript 라이브러리란?

JavaScript 라이브러리는 웹 개발자가 표준 JavaScript 함수를 실행하기 위해 재사용할 수 있는 미리 작성된 코드 조각의 모음입니다. JavaScript 라이브러리 코드는 필요에 따라 나머지 프로젝트 코드에 연결됩니다. JavaScript 애플리케이션 코드를 집이라고 가정한다면, JavaScript 라이브러리는 개발자가 집의 기능을 개선하는 데 사용할 수 있는 미리 제작된 가구와 같습니다.

### 3) 자바스크립트 특징

- ① 자바스크립트는 객체기반 언어이며, HTML 문서 내에 내장되어 프로그래밍 요소를 추가할 수 있습니다.
- ② 모든 객체는 숨겨진 링크인 프로토타입을 가지고 있습니다.
- ③ 인터프리터 언어로서 클라이언트 웹 브라우저에서 실행됩니다.
- ④ 컨텍스트를 만들고, 자신의 Scope : 클로저를 구현하여 실행합니다.
- ⑤ Node.js 프레임워크를 이용해 백엔드 개발에서도 사용 가능합니다.
- ⑥ 객체형 / 함수형 프로그래밍 모두 표현 가능합니다.

### 4) 자바스크립트 장단점

#### - 장점

HTML 파일 내에서 작성하고 바로 작동시켜 볼 수 있기 때문에, 개발 속도가 빠릅니다. js는 웹 브라우저에서 동작하기 때문에 운영체제의 제한을 받지 않으며, 여러 환경에서 개발이 가능합니다. 또한, NodeJS 를 활용하면, 프론트엔드 & 백엔드 모두 개발 가능합니다.

#### - 단점

웹 브라우저 상에 소스 코드가 노출되어 있어, 공개가 되기 때문에 보안에 취약합니다. js는 컴파일하지 않아도 되기 때문에, 복사하여 그대로 사용할 수 있습니다. 마지막으로, 한정된 객체, 객체

함수를 제공하기 때문에 개발시에 오류에 대한 여러 사항을 고려할 필요가 있습니다.

#### IV. React

##### 1) React 란?

React.js는 특히 SPA(Single Page Application)을 위한 사용자 인터페이스를 구축하는 데 사용되는 오픈 소스 JavaScript 라이브러리입니다. React는 Facebook에서 일하는 소프트웨어 엔지니어 Jordan Walke가 처음 만들었습니다. 웹 및 모바일 앱의 view layer를 처리하는 데 사용됩니다. 데이터 바인딩 기법을 통해서 여러가지 정보를 한 곳에 모아서 보여줄 수 있고, 뷰포트 개념을 도입했기 때문에 스크롤링 효과 등 시각적인 부분에서의 표현이 자유롭습니다. SPA (Single Page Application)방식이라 페이지 이동없이 빠르게 콘텐츠를 제공할 수 있다는 것도 매력적인 점입니다. 또한 React를 사용하면 재사용 가능한 UI 구성 요소를 만들 수 있습니다. React를 사용하면 개발자가 페이지를 다시 로드하지 않고도 데이터를 변경할 수 있는 대규모 웹 애플리케이션을 만들 수 있습니다.

##### 2) React 특징

- React는 선언형입니다.  
선언형이란 소프트웨어 공학에서 자주 접하게 되는 개념은 패러다임(Paradigm)입니다. 즉, 프로그래밍을 어떻게 할 것인가 하는 생각에 대한 논의가 활발하게 이루어집니다. 이와 같은 선언형 특성은, 리액트를 사용할 때 화면 설계라는 초점에 맞춰서 개발할 수 있도록 해주므로, 다른 부분에 대한 고민을 최소화 해주어 높은 생산성을 보장할 수 있도록 해줍니다.
- React는 컴포넌트 기반으로 재사용성이 뛰어납니다.  
컴포넌트는 독립적인 단위의 소프트웨어 모듈을 말합니다. 즉 소프트웨어를 독립적인 하나의 부품으로 만드는 방법입니다. 리액트는 웹에서 쓰는 각 요소들을 컴포넌트로 만들 수 있게 해 기존의 UI를 다른 화면에서 다시 쓰거나, 다른 프로젝트에서 다시 쓸 수 있도록 하는 장점(높은 재사용성)을 가집니다.
- React는 Virtual DOM(가상돔)기반으로 가볍습니다.  
웹 브라우저와 자바스크립트를 동작시키기 위한 V8엔진도 소프트웨어입니다. 즉, 이들이 실행되면 메모리(RAM)가 할당이 됩니다. 가상돔은 이 메모리 단에서 컴포넌트에 대한 정보를 생성하고 비교하여 전체 DOM 트리가 업데이트가 필요한 경우 이를 반영하는 방식을 말합니다.
- React 컴포넌트는 state와 props 을 가집니다.  
리액트는 내부적으로 State와 Props를 가집니다. 이는, UI(User Interface)가 사용자의 동작(Actions)에 따라 다른 UI나 Action이 필요하기 때문입니다.

#### V. Angular

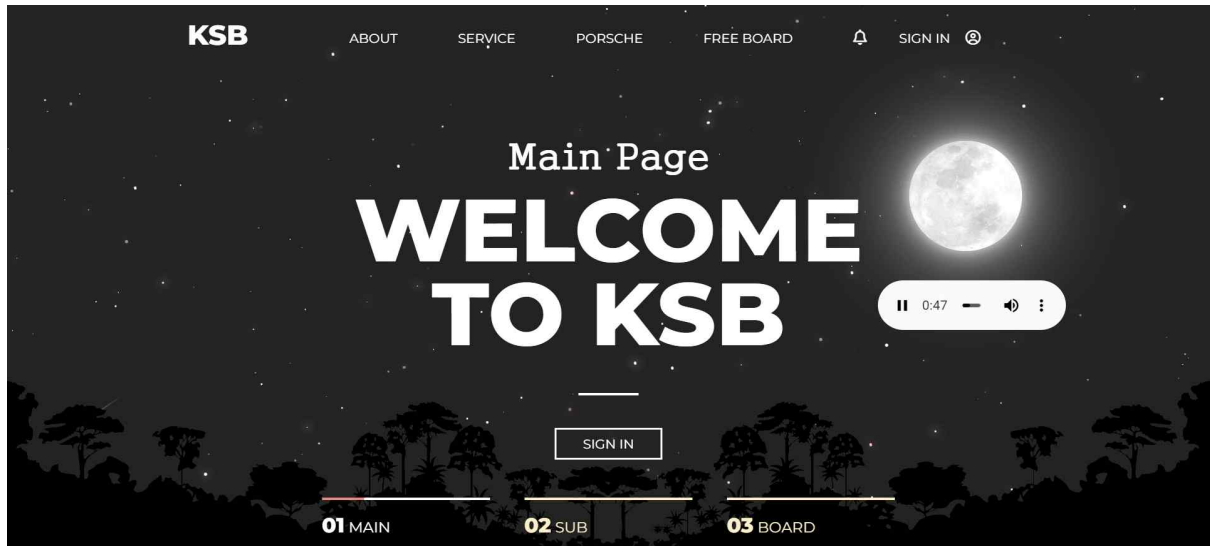
##### 1) Angular란?

Angular란 Google에서 만든 SPA 방식의 프론트엔드 개발을 위한 자바스크립트 프레임워크입니다. 웹 어플리케이션은 물론 모바일 웹, 네이트브 모바일과 데스크탑 어플리케이션까지 프론트엔드 개발에 필요한 대부분의 기능을 갖추고 있습니다. 사용언어는 TypeScript 혹은 ES6을 사용할 수 있지만, 개발그룹측에서는 TypeScript를 사용할 것을 더욱 권장하고 있습니다. 실제 관련 문서도 TypeScript를 기준으로 설명 되어있는 것이 더 풍부합니다.  
또한 Angular를 이용하여 개발을 하다보면 Spring MVC와 굉장히 유사한 방식으로 프로그래밍이 진행된다는 것을 느낄 수 있습니다.

##### 2) Angular의 특징

- 문서 객체 모델  
DOM(Document Object Model)은 XML 또는 HTML 문서를 각 노드가 문서의 일부를 나타내는 트리 구조로 취급합니다. Angular는 일반 DOM을 사용합니다. 이미 업데이트된 것을 업데이트하는 대신 Angular는 HTML 태그의 전체 트리 구조를 업데이트합니다.
- 타입스크립트  
TypeScript는 사용자가 이해하기 쉬운 JavaScript 코드를 작성할 수 있도록 JavaScript에 유형 집합을 정의합니다. 모든 TypeScript 코드는 JavaScript로 컴파일되며 모든 플랫폼에서 원활하게 실행할 수 있습니다. TypeScript는 Angular 애플리케이션 개발에 필수적입니다. 그러나 더 나은 구조를 제공하는 동시에 코드베이스를 더 쉽게 이해하고 유지 관리할 수 있으므로 적극 권장합니다.
- 데이터 바인딩 - 테스트

## Index.html



### 메인페이지 (php)

nav태그를 사용하여 위에 상단바를 구성하였습니다. 상단바에서는 로그인, 메인홈화면, 창작페이지 슬라이드, 게시판으로 갈 수 있게 해줍니다. 메인홈화면의 슬라이드는 7초마다 자동으로 넘어갈 수 있으며, 밑에 상태바를 클릭해도 원하는 슬라이드로 넘어갈 수 있게 해주었습니다. 배경화면은 이미지를 가져와 슬라이드 및 div가 움직여도 고정되게 해주었고 360도 회전이 가능한 별과 가끔 별뿔별이 떨어지게 하여 예쁜 밤하늘로 보일 수 있게 구현하였습니다.

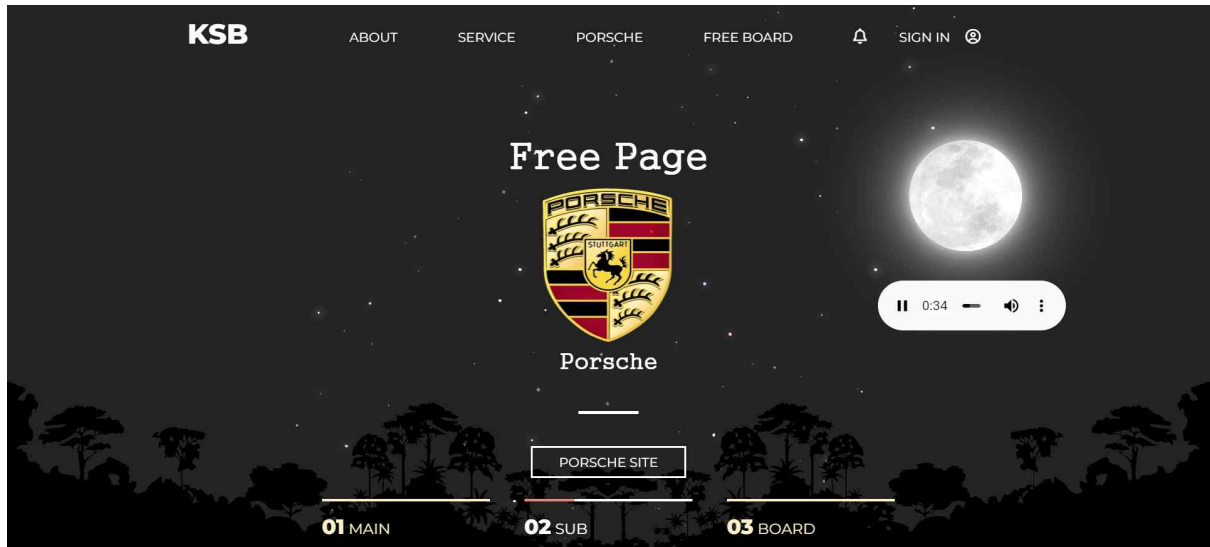
```
function bannerSwitcher() {
    next = $('.sec-1-input').filter(':checked').next('.sec-1-input');
    if (next.length) next.prop('checked', true);
    else $('.sec-1-input').first().prop('checked', true);
}

var bannerTimer = setInterval(bannerSwitcher, 7000);

$('.nav .controls label').click(function() {
    clearInterval(bannerTimer);
    bannerTimer = setInterval(bannerSwitcher, 7000);
});
```

처음 메인홈화면에 갔을 때 Welcome 부분부터 슬라이드가 보일 수 있도록 체크 기능을 사용하여 보이게 해줍니다. 슬라이드는 자동으로 7초마다 각 슬라이드가 다음 슬라이드로 넘어갈 수 있게 구현하였습니다. 뿐만 아니라, 하단 상태바를 클릭하면 클릭한 원하는 슬라이드로 갈 수 있도록 클릭 이벤트 함수도 만들어 구현하였습니다.

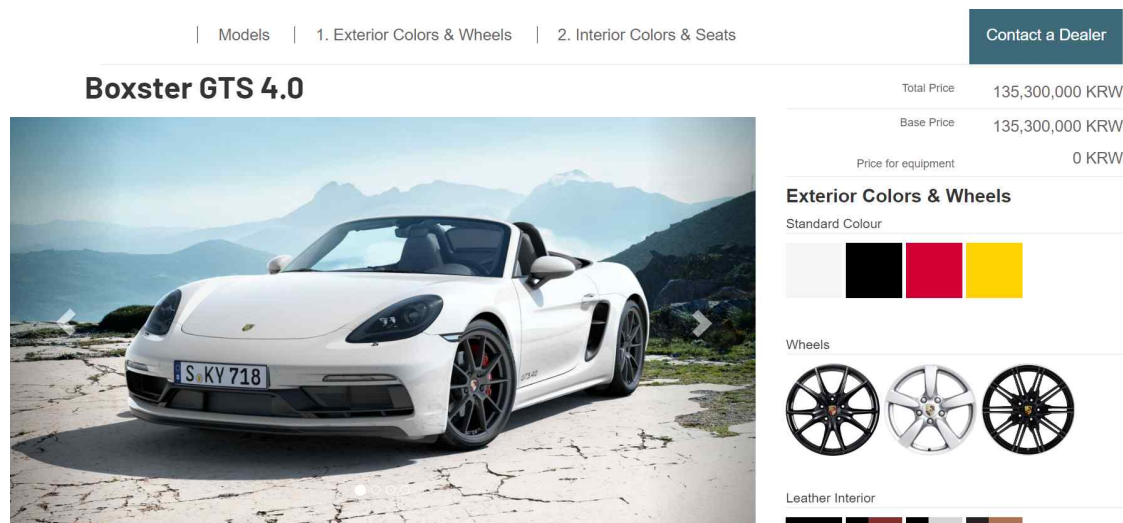
### 창작 페이지



메인페이지의 SUB 상태바에서 porsche site 버튼을 클릭하면 제가 만든 창작 페이지인 포르쉐 페이지로 넘어가게 됩니다.



포르쉐 사이트는 실제 포르쉐 사이트와 유사하게 만들어보았습니다. 이 페이지에서는 포르쉐 사이트에서 동영상을 다운받아 배경에 동영상을 넣고 자동 실행 되도록 하였으며 '나만의 포르쉐 만들기' 버튼을 누르면 나만의 포르쉐 차를 꾸밀 수 있는 사이트로 넘어가도록 구현하였습니다.



나만의 포르쉐를 만들 수 있는 사이트입니다. 왼쪽 차 사진은 4개의 사진으로 구성되어 있는데 앞, 뒤, 내부 앞, 내부 뒤로 구성되어 있습니다. 오른쪽 사이드바에는 차 색, 바퀴 종류, 시트 색이 있는데 이것들을 클릭하면 왼쪽 차 사진들의 디자인이 바뀌게 됩니다. 오른쪽 사이드바에서 원하는 차로 외형을 바꾼 뒤 'Are you ready?'란 버튼을 누르면 내가 만든 차의 결과가 나오는 사이트로 넘어갑니다.

```
var carOutImg1 = document.querySelector('#outimg1');
var carOutImg2 = document.querySelector('#outimg2');
var carInImg1 = document.querySelector('#inimg1');
var carInImg2 = document.querySelector('#inimg2');

// var carColor = "white";
// var carWheel = "basewheel";
// var carSeat = "blackseat";

let carColor = ['white', 'black', 'red', 'yellow'];
let carWheel = ['basewheel', 'carrerawheel', 'turbowheel'];
let carSeat = ['blackseat', 'redseat', 'whiteseat', 'brownseat'];

var carColorList = document.querySelectorAll('.carC');
for(let i = 0; i < carColorList.length; i++){
  carColorList[i].addEventListener('click', changeImg1);
}
```

위처럼 클릭하면 그 값들을 받아온 후 배열에서 클릭한것의 값을 가져온 뒤 이미지를 가져오게 됩니다.

```
function a(indexCarColor, indexCarWheels, indexCarSeats){
  carOutImg1.setAttribute('src', '/images/porsche_'+carColor[indexCarColor]+'_'+carWheel[indexCarWheels]+'_'+carSeat[indexCarSeats]+'_jpg');
  carOutImg2.setAttribute('src', '/images/porsche_back_'+carColor[indexCarColor]+'_'+carWheel[indexCarWheels]+'_blackseat.jpg');
  carInImg1.setAttribute('src', '/images/porsche_internal_'+carSeat[indexCarSeats]+'_jpg');
  carInImg2.setAttribute('src', '/images/porsche_seats_'+carSeat[indexCarSeats]+'_jpg');
}
```

이런식으로 값들로 사진들의 이미지를 가져옵니다. 사진들의 이름은 미리 변수들이 들어갈 수 있게 맞춰놓습니다.



## Your Customized Carl



▲ If you want to buy it? Hover your mouth over the image!

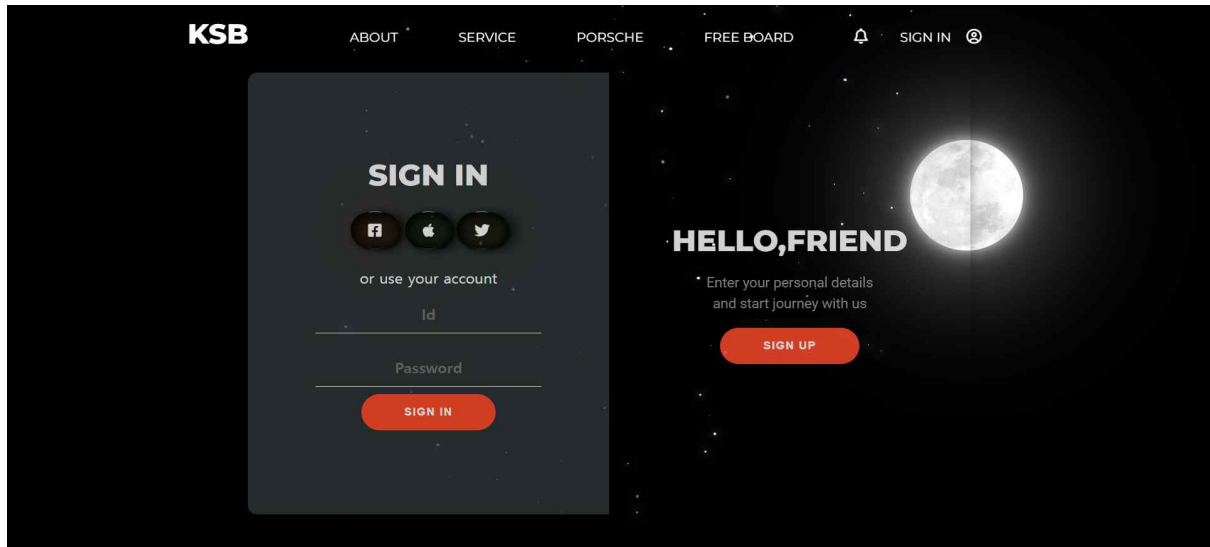
RETRY

사용자가 커스텀한 차의 결과가 나오는 사이트입니다. 변수값들로 사진을 가져온 후 저사진에 마우스 온을 하면 'Buy Me Now!' 란 문구가 나타납니다. 그리고 Retry 버튼을 누르면 다시 나만의 차를 커스텀 할 수 있는 사이트로 돌아가게 됩니다.

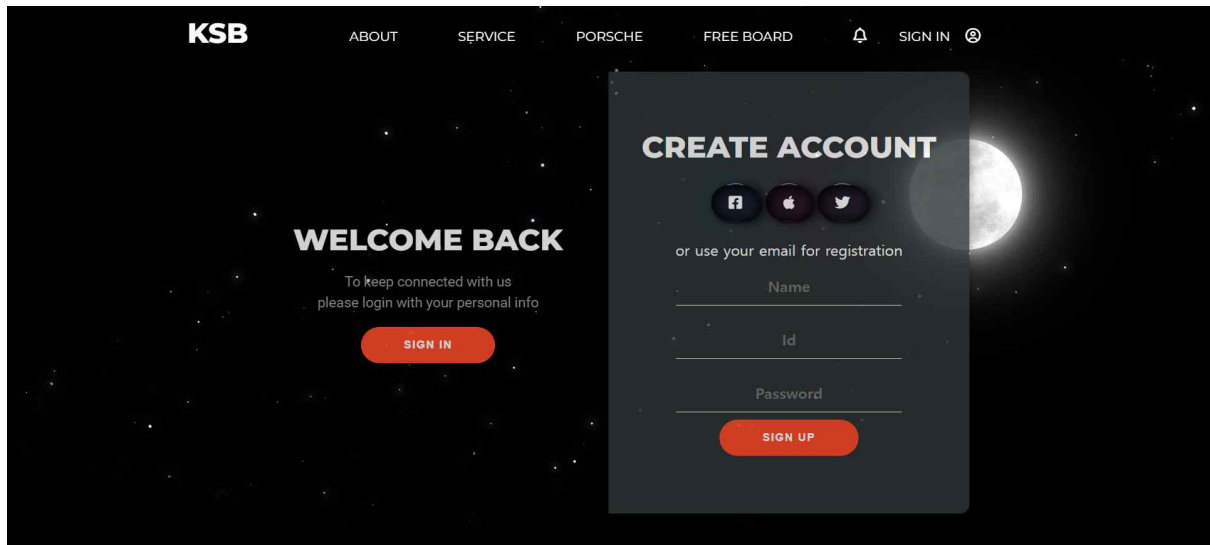
```
.card {  
  /* background: #191c29; */  
  background-color: white;  
  background-image: url('/images/white.jpg');  
  background-repeat: no-repeat;  
  background-position: center center;  
  background-size: 250% 250%;  
  
  width: var(--card-width);  
  height: var(--card-height);  
  padding: 3px;  
  position: relative;  
  border-radius: 6px;  
  justify-content: center;  
  align-items: center;  
  text-align: center;  
  display: flex;  
  /* font-size: 1.5em; */  
  font-size: 2.8em;  
  color: rgb(88 199 250 / 0%);  
  cursor: pointer;  
  font-family: cursive;  
}
```

이게 차의 결과가 나오는 css 코드입니다.

로그인



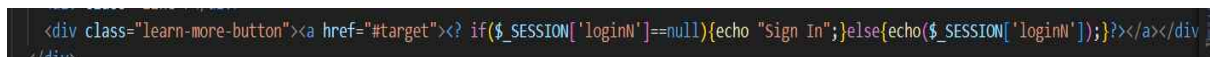
로그인을 할 수 있는 곳입니다. 회원가입을 한 후 저장된 아이디와 비밀번호를 입력하여 로그인 할 수 있습니다.



회원가입을 할 수 있는 곳입니다. DB와 연동하여 입력한 이름, 아이디, 비밀번호를 저장할 수 있도록 해주었습니다.



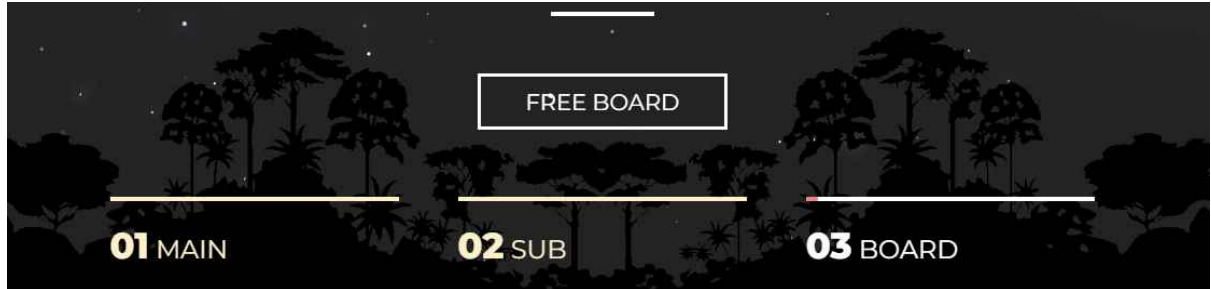
로그인을 한 후 나브바에 로그인한 사용자의 이름과 sign out 문구가 나오게 구현을 해보았습니다. sign out을 누르면 로그아웃이 됩니다.



이렇게 처음엔 디폴트 값이 sign in 이었고(불러온 값이 null 이어야됨) 로그인을 하게되면 값이 null이 아니게 되니까 사용자 이름과 sign in 이 sign out으로 바뀌게 됩니다.

슬라이드바



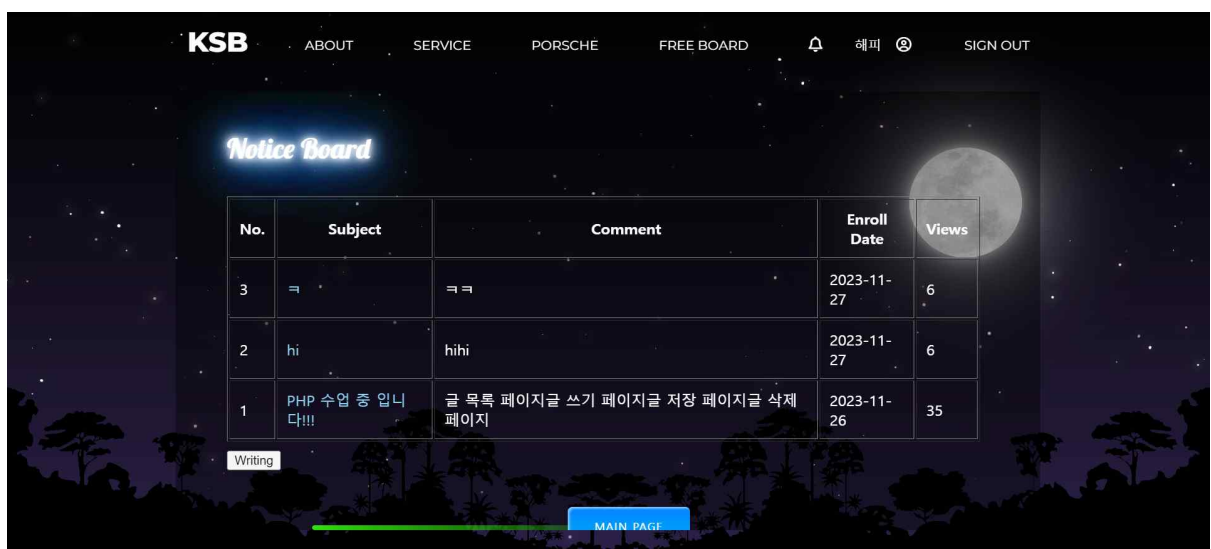


슬라이드가 7초뒤에 자동으로 다음 슬라이드로 넘어가게 됩니다. 프로세스바의 상태변화 또한 7초간 상태바가 모두 빨간색으로 채워지게 구현을 했습니다. 클릭하면 원하는 슬라이드로 또한 넘어갈 수 있습니다.



포르쉐 창작 페이지의 차가 나오는 사진 부분도 슬라이드로 구현했습니다. 이 또한 일정시간이 지나면 다음 사진으로 자동으로 넘어갈 수 있게 하였으며 원하는 순서의 사진이나 이전, 다음 버튼을 누르면 이전, 이후 사진으로 갈 수 있게 하였습니다.

자유게시판(삽입, 삭제, 수정 등)



자유 게시판입니다. 자유 게시판 또한 로그인 페이지와 같이 홈화면과 연결되어있습니다. 일단 자유게시판임을 알리는 notice board 문구를 네온 글씨가 나타나게 해주었습니다. 원래 만들어둔 게시판을 php로 include 해서 미리 위치를 설정해준 div안에 나타나게(보여지게) 구현하였습니다.

게시판은 수정, 삭제, 글쓰기, 내용보기가 가능하며 만약 자유글이 많아서 div를 overflow할 시, 스크롤 해서 볼 수 있게 구현하였습니다.