|  |  |
| --- | --- |
| 과목명 | 웹프로그래밍 |
| 담당교수 | 천인국 |
| 학과 | 컴퓨터공학과 |
| 학년 | 2 |
| 학번 | 20204101 |
| 이름 | 신동욱 |
| 제출일 | 11.26 |

로고, 등록 상표, 폰트, 상징이(가) 표시된 사진

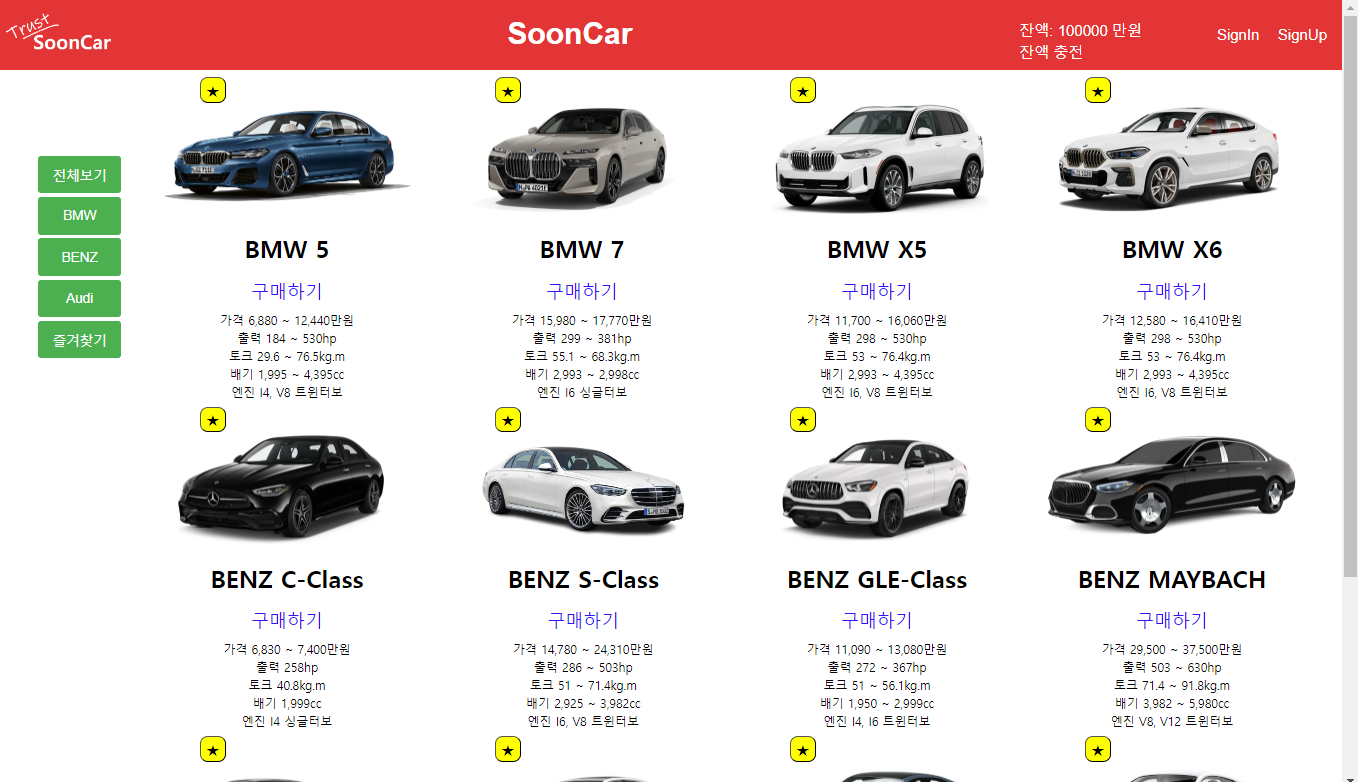
자동 생성된 설명

**웹프로그래밍**

**홈쇼핑 with JS**

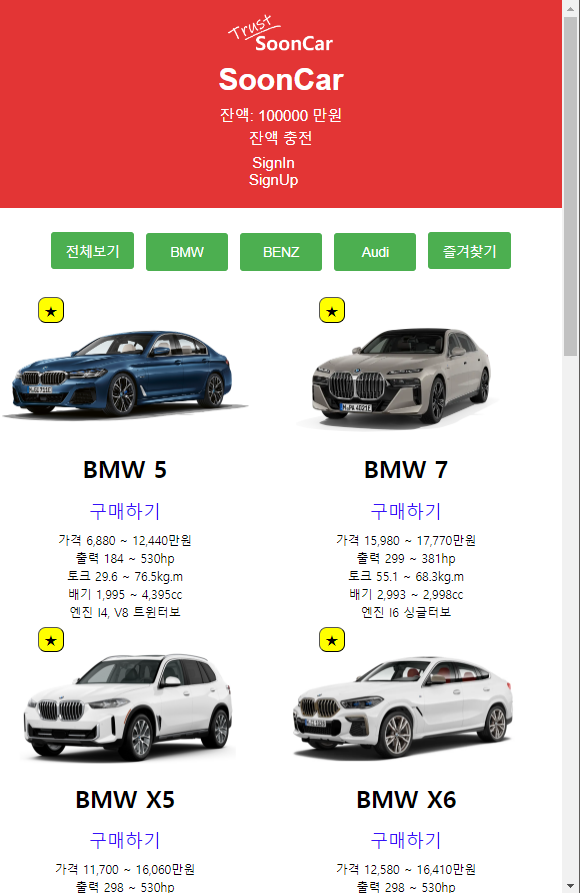
1. 추가한 기능
   1. 반응형 웹 디자인
   2. 특정 항목의 아이템만 표시
   3. 즐겨찾기
   4. 구매 및 잔액 차감
   5. 잔액 충전
   6. 로그인 회원가입 폼 유효성 검사
2. 소스코드 및 설명

**1. 추가한 기능**

****

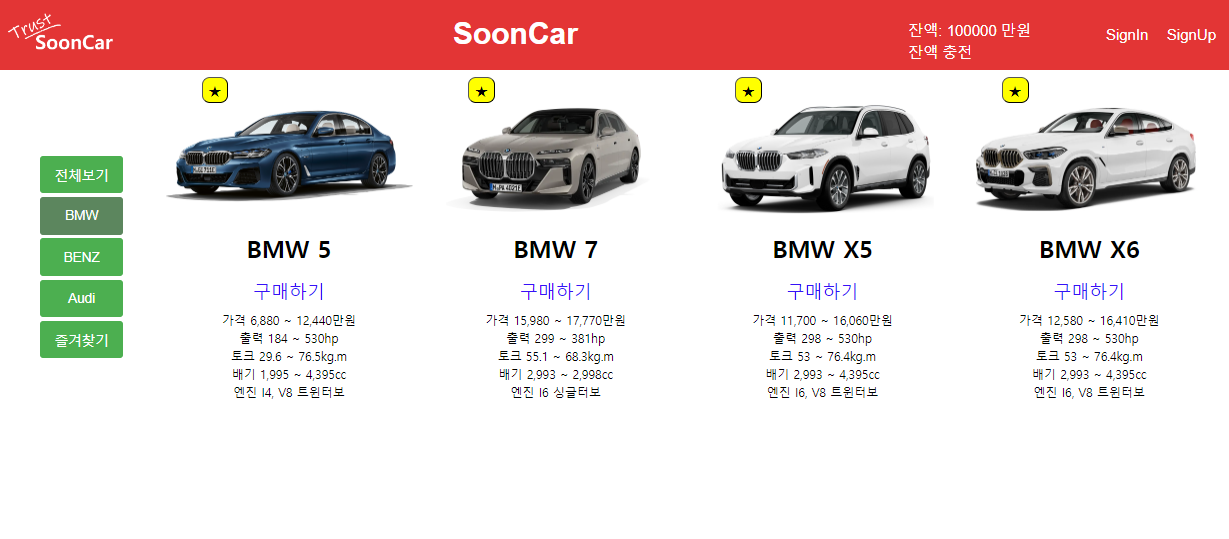
초기 화면이자 메인 화면

**추가한 기능 1. 반응형 웹 디자인**

중단점을 지정하여 화면이 일정 크기 이하보다 작아진다면 레이아웃을 달리 구성했다

네비게이션 바에 있는 잔액, 잔액 충전, SingIn, SingUp 버튼을 세로로 배치하였다

또한 특정 목록만을 표시하는 버튼들을 가로로 나열하였다  
**추가한 기능 2. 특정 항목의 아이템만 표시**

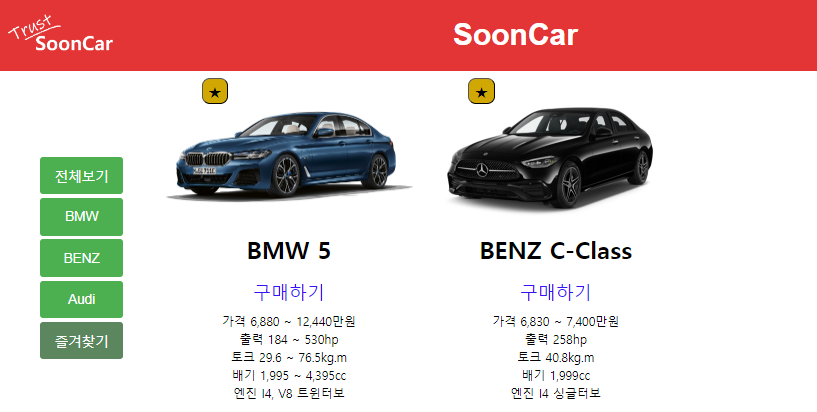


좌측의 초록색 BMW 버튼을 누른다면 특정 브랜드의 요소만 화면에 표시된다

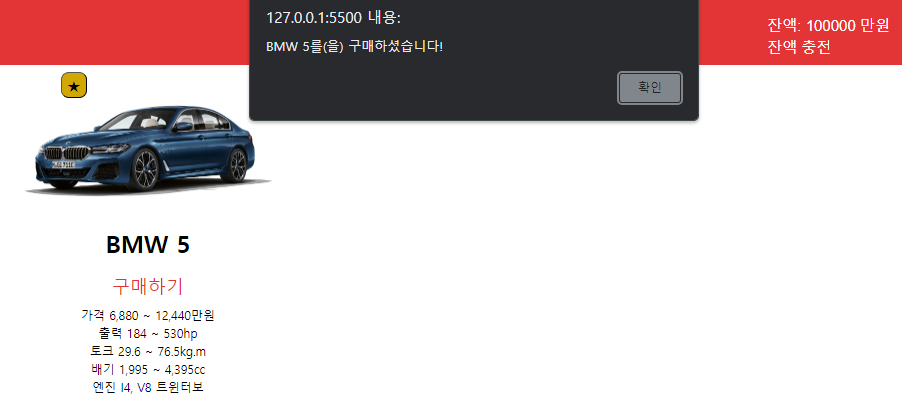
**추가한 기능 3. 즐겨찾기**

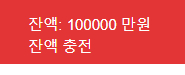
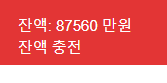
 

각 요소 사진 위의 별 버튼을 누르면 즐겨찾기가 되어 버튼의 디자인이 바뀐다

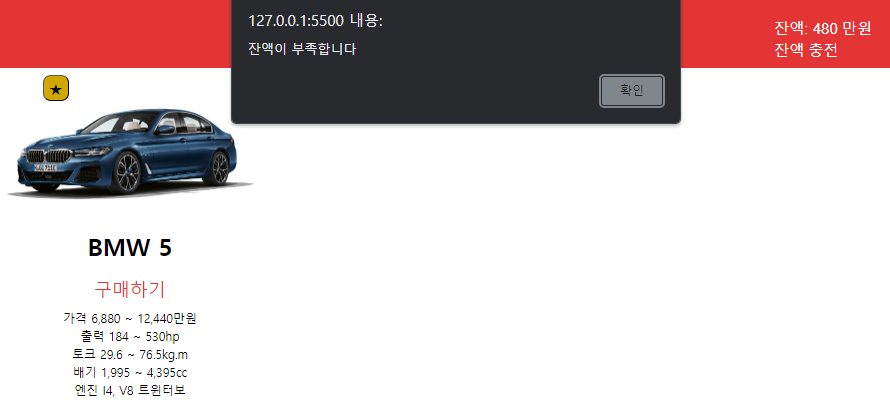
   
**추가한 기능 4. 구매 및 잔액 차감**

추가한 기능 2. 에 확장하여, 좌측의 초록색 즐겨찾기 버튼을 누르면 즐겨찾기를 한   
항목들이 표시된다



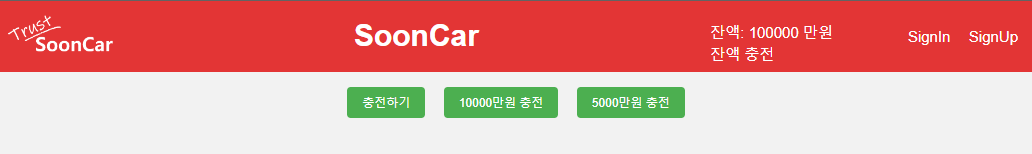
자동차의 구매하기 버튼을 클릭하현 구매 메시지가 출력된다  
 

이후 우측 상단의 잔액이 차감된다

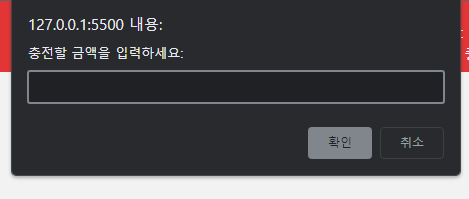
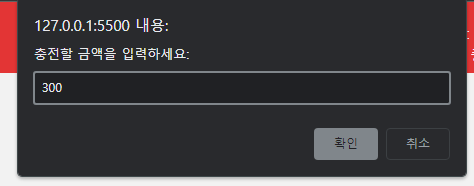


만약 잔액이 모자라다면 구매에 실패한다

**추가한 기능 5. 잔액 충전**

  
우측 상단의 잔액 충전 버튼을 누르면 잔액 충전 페이지로 이동한다

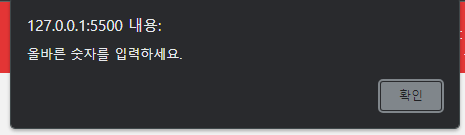
잔액을 충전할 수 있는 버튼들이 표시된다


버튼 “충전하기”를 클릭하면 사용자에게 금액을 입력 받는다.

사용자가 충전할 금액을 입력하면

잔액에 더해진다



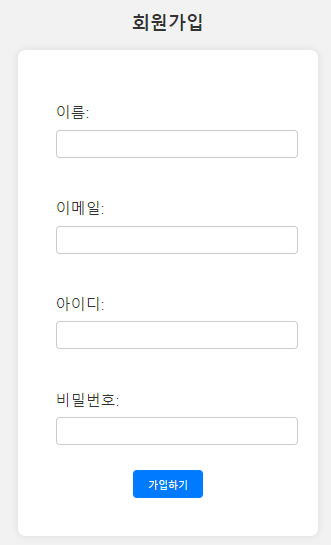
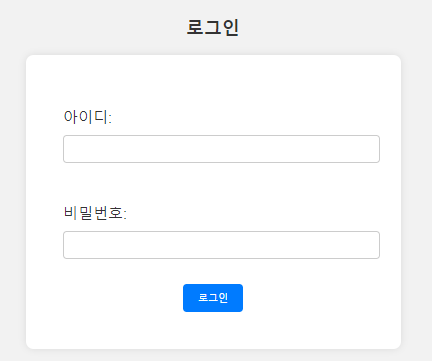
만약 유효하지 않은 값을 입력하면 충전이 되지 않는다



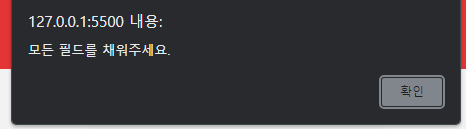
이 버튼을 누르면 사용자에게 입력을 받지 않아도 정해진 값만큼 잔액에 더해진다



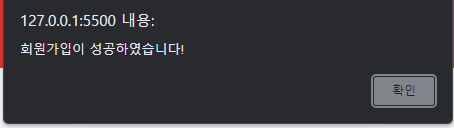
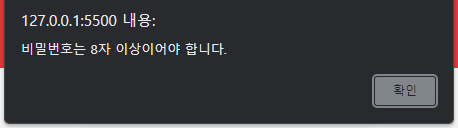
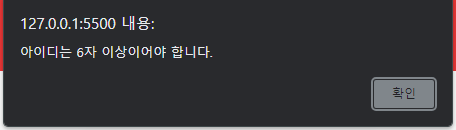
**추가한 기능 6. 로그인 회원가입 폼 유효성 검사**



우측 상단의 SignIn, SingUp 버튼을 클릭하면 로그인, 회원가입 페이지로 이동한다.



필드에 공백이 있는지 검사한다



모든 조건을 만족한다면 성공 메시지 출력

비밀번호가 8자 이상인지 검사한다

아이지가 6자 이상인지 검사한다

**2. 소스코드 및 설명**

**텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명** 전체 경로

**navbar.html**

모든 html 파일은 navbar.html을 포함한다

<!DOCTYPE html>

<html>

<head>

뷰포트의 너비를 장치 너비로 설정한다

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>순카</title>

    <style>

        @media screen and (max-width: 800px) {

            .header-content {

                display: inline;

중단점을 설정한다  
  
이하의 요소들의 display를 block으로 만든다  
글자 정렬을 가운데로 모은다

                max-width: 100%;

            }

            ul li {

                display: block;

            }

            ul {

                text-align: center;

                margin-left: 0;

                padding: 0;

            }

            .money-box {

                text-align: center;

            }

            .header-title {

                text-align: center;

            }

            .logo {

                text-align: center;

            }

        }

    </style>

</head>

<body>

    <header>

        <div class="header-content">

            <div class="logo">

                <img  src="image/logo.png" alt="SoonCar Logo">

            </div>

            <div class="header-title">

                <a href="main.html" target="\_parent">SoonCar</a>

            </div>

id를 지정한다. 이 부분은 함수로 인해 값이 변한다

            <div class="money-box">

                <p>잔액: <span class="money" id="moneyValue">100000</span> 만원</p>

                <a href="chargemoney.html" target="\_parent">잔액 충전</a>

            </div>

            <ul>

                <li><a href="signin.html" target="\_parent">SignIn</a></li>

                <li><a href="signup.html" target="\_parent">SignUp</a></li>

            </ul>

        </div>

    </header>

돈을 소모시키는 함수  
moneyValue의 값을 가져온다

<script>

    function spendMoney(model, money) {

        let currentMoney = parseInt(document.getElementById('moneyValue').textContent);

        if (money > currentMoney) {

인수로 들어온 money가 현재 돈보다 더 크면  
함수 종료

            alert("잔액이 부족합니다");

            return;

        }

돈 갱신

        let newMoney = currentMoney - money;

        document.getElementById('moneyValue').textContent = newMoney;

        alert(model + "를(을) 구매하셨습니다!");

    };

돈을 소모시키는 함수  
moneyValue의 값을 가져온다

    function chargeMoney(money) {

        let currentMoney = parseInt(document.getElementById('moneyValue').textContent);

        let newMoney = currentMoney + money;

        document.getElementById('moneyValue').textContent = newMoney;

    };

돈 갱신

</script>

</body>

</html>

**signin.html**

<!DOCTYPE html>

<html>

<head>

뷰포트의 너비를 장치 너비로 설정한다

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <style>

        @media screen and (max-width: 800px) {

            .nav iframe {

중단점을 설정한다  
  
네비게이션 바가 들어있는 iframe의 크기를 조절한다

                width: 100%;

                height: 300px;

                border: none;

            }

        }

    </style>

</head>

<body>

    <div class="nav">

        <iframe src="navbar.html" frameborder="0" scrolling="no"></iframe>

    </div>

폼에서 submit 버튼을 누르면 vaildateForm   
함수를 호출한다

    <h2>로그인</h2>

    <form onsubmit="validateForm()">

        <p>아이디:</p>

        <input type="text" id="userid"><br><br>

        <p>비밀번호:</p>

        <input type="password" id="password"><br><br>

        <input type="submit" value="로그인">

    </form>

    <script>

입력 양식 유효성 검사

        function validateForm() {

            var userid = document.getElementById('userid').value;

            var password = document.getElementById('password').value;

            if (userid === '' || password === '') {

                alert('모든 필드를 채워주세요.');

아이디와 비밀번호가 비어있으면 false 반환

                return false;

            }

            alert('로그인 성공!');

그렇지 않다면 true 반환

            return true;

        }

    </script>

</body>

</html>

**signup.html**

<!DOCTYPE html>

<html>

<head>

뷰포트의 너비를 장치 너비로 설정한다

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <style>

        @media screen and (max-width: 800px) {

            .nav iframe {

중단점을 설정한다  
  
네비게이션 바가 들어있는 iframe의 크기를 조절한다

                width: 100%;

                height: 300px;

                border: none;

            }

        }

    </style>

</head>

<body>

    <div class="nav">

        <iframe src="navbar.html" frameborder="0" scrolling="no"></iframe>

    </div>

폼에서 submit 버튼을 누르면 vaildateForm   
함수를 호출한다

    <h2>회원가입</h2>

    <form onsubmit="validateForm()">

        <p>이름:</p>

        <input type="text" id="name"><br><br>

        <p>이메일:</p>

        <input type="email" id="email"><br><br>

        <p>아이디:</p>

        <input type="text" id="userid"><br><br>

        <p>비밀번호:</p>

        <input type="password" id="password"><br><br>

        <input type="submit" value="가입하기">

    </form>

    <script>

입력 양식 유효성 검사

        function validateForm() {

            var name = document.getElementById('name').value;

            var email = document.getElementById('email').value;

            var userid = document.getElementById('userid').value;

            var password = document.getElementById('password').value;

            if (name === '' || email === '' || userid === '' || password === '') {

                alert('모든 필드를 채워주세요.');

                return false;

필드가 비어있는지 검사

            }

            if (userid.length < 6) {

아이디 길이 검사

                alert('아이디는 6자 이상이어야 합니다.');

                return false;

            }

비밀번호 길이 검사

            if (password.length < 8) {

                alert('비밀번호는 8자 이상이어야 합니다.');

                return false;

            }

            alert('회원가입이 성공하였습니다!');

            return true;

그렇지 않다면 true 반환

        }

    </script>

</body>

</html>

**chargemoney.html**

돈을 충전하는 html 파일

<!DOCTYPE html>

<html>

<head>

뷰포트의 너비를 장치 너비로 설정한다

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <style>

        @media screen and (max-width: 800px) {

            .nav iframe {

중단점을 설정한다  
  
네비게이션 바가 들어있는 iframe의 크기를 조절한다

                width: 100%;

                height: 300px;

                border: none;

            }

        }

    </style>

</head>

<body>

    <div class="nav">

        <iframe src="navbar.html" id="navFrame"></iframe>

    </div>

버튼을 누르면 charge 함수가 호출된다

    <button onclick="charge(0)">충전하기</button>

    <button onclick="charge(10000)">10000만원 충전</button>

    <button onclick="charge(5000)">5000만원 충전</button>

    <script>

돈을 충전하는 함수  
들어온 money 인수가 0이라면 사용자에게 돈을 입력 받는다

        function charge(money) {

            if (money == 0) {

                var amount = parseInt(prompt("충전할 금액을 입력하세요:"));

                if (!isNaN(amount)) {

                    var moneyBoxFrame = document.getElementById('navFrame');

                    moneyBoxFrame.contentWindow.chargeMoney(amount);

                } else {

올바른 숫자가 입력됐다면 chargeMoney 함수를 호출한다 chargeMoney는 navbar.html에 있다

                    alert("올바른 숫자를 입력하세요.");

                }

            }

            var moneyBoxFrame = document.getElementById('navFrame');

            moneyBoxFrame.contentWindow.chargeMoney(money);

        }

    </script>

</body>

</html>

**main.html**

<!DOCTYPE html>

<html>

<head>

뷰포트의 너비를 장치 너비로 설정한다

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width initial-scale=1.0">

    <title>순카</title>

    <style>

중단점을 설정한다  
  
네비게이션 바가 들어있는 iframe의 크기를 조절한다  
버튼들의 크기와 배치를 조절한다

        @media screen and (max-width: 800px) {

            .car-container {

                display: grid;

                grid-template-columns: 1fr 1fr;

            }

            .nav iframe {

                width: 100%;

                height: 300px;

                border: none;

            }

            .filter-button {

                display: inline;

            }

            .filter-buttons {

                margin-top: 0;

                float: none;

                margin-bottom: 30px;

            }

        }

    </style>

</head>

<body>

    <div class="nav">

        <iframe src="navbar.html" id="navFrame"></iframe>

    </div>

<div class="filter-buttons">

        <button class="filter-button" onclick="showAll()">전체보기</button>

        <button class="filter-button" onclick="filterByBrand('BMW')">BMW</button>

        <button class="filter-button" onclick="filterByBrand('BENZ')">BENZ</button>

        <button class="filter-button" onclick="filterByBrand('Audi')">Audi</button>

        <button class="filter-button" onclick="filterStarredCar()">즐겨찾기</button>

전체보기 버튼을 누르면 showAll 함수를 호출한다  
브랜드 버튼을 누르면 filterByBrand 함수를 호출한다  
즐겨찾기 버튼을 누르면 filterStarredCar 함수를 호출한다

</div>

<div class="car-container">

        <div class="car-item">

            <button class="star" onclick="toggleStar(this, 'BMW 5')">★</button>

            <img src="image/BMW5.png" alt="BMW5.png">

한 개의 car-item에는 별 버튼을 포함한다  
별 버튼을 누르면 toggleStar 함수를 호출한다

            <div class="car-info">

                <p class="car-title">

                    BMW 5

                </p>

                <button class="car-price" onclick="purchase('BMW 5', 12440)">구매하기</button>

구매하기 버튼을 누르면 purchase함수를 호출한다

                <p>가격 6,880 ~ 12,440만원</p>

                <p>출력 184 ~ 530hp</p>

                <p>토크 29.6 ~ 76.5kg.m</p>

                <p>배기 1,995 ~ 4,395cc</p>

                <p>엔진 I4, V8 트윈터보</p>

            </div>

        </div>

이하 요소들은 생략

showAll 함수를 호출하면 모든 car-item 요소들을 순회하여 보이게 한다

<script>

    function showAll() {

        var carItems = document.querySelectorAll('.car-item');

        carItems.forEach(function (item) {

            item.style.display = '';

        });

    }

    function filterByBrand(brand) {

filterByBrand 함수를 호출하면 모든 car-item 요소들을 순회하여 car-title에 있는 글자를 추출한다  
글자를 추출해서 글자의 앞 단어를 추출하여 인수와 동일하면 화면에 보이게 한다

        showAll()

        var carItems = document.querySelectorAll('.car-item');

        carItems.forEach(function (item) {

            var carTitleText = item.querySelector('.car-title').innerText;

            var carBrand = carTitleText.split(' ')[0].toUpperCase();

            if (carBrand === brand.toUpperCase()) {

                item.style.display = '';

            } else {

                item.style.display = 'none';

            }

        });

    }

purchase 함수를 호출하면 navbar.html에 있는 spendMoney 함수를 호출한다

    function purchase(model, price) {

        var moneyBoxFrame = document.getElementById('navFrame');

        moneyBoxFrame.contentWindow.spendMoney(model, price)

    };

    var starredCar = [];

toggleStar 함수를 호출하면 버튼의 현재 상태를 파악한다

    function toggleStar(button, model) {

        // 버튼의 현재 상태를 확인하여 토글

        var isStarred = button.classList.contains('starred');

        // 토글된 상태에 따라 스타일 변경

        if (isStarred) {

            button.classList.remove('starred');

토글이 해제된 상태면 리스트에서 요소를 삭제한다

            var valueToRemove = model;

            var index = starredCar.indexOf(valueToRemove);

            if (index !== -1)

                starredCar.splice(index, 1);

            console.log(starredCar);

        } else {

그렇지 않으면 리스트에 요소를 추가한다

            button.classList.add('starred');

            starredCar.push(model);

            console.log(starredCar);

        }

    }

filterStarredCar 함수를 호출하면 리스트에 있는 요소를 화면에 보이게 한다

    function filterStarredCar() {

        var carItems = document.querySelectorAll('.car-item');

        carItems.forEach(function (item) {

            var model = item.querySelector('.car-title').innerText;

            console.log(model);

            var isStarred = starredCar.includes(model);

            if (isStarred) {

                item.style.display = ''

            } else {

                item.style.display = 'none';

            }

        });

    }

</script>

</body>

</html>