

JEGYZŐKÖNYV

Web Technológiák 1.

Féléves feladat

Sportautók – interaktív prezentáció

Készítette: **Orosz Zalán Zétény**

Neptun kód: **XPUYJX**

Dátum: 2025. december

Miskolc, 2025

Tartalomjegyzék

1. Bevezetés	2
2. Feladat leírása	2
3. Weblap felépítése	2
3.1. Fejléc és főcím	2
3.2. Swiper carousel	2
3.3. Interaktív elemek	2
4. Scriptek	3
4.1. Autók adatai objektumban	3
4.2. Swiper inicializálása	3
4.3. Dinamikus cím és logó váltás	3
4.4. EXPLORE gomb működése	3
4.5. Műszaki adatok megjelenítése	4
4.6. Videó overlay kezelése	4
4.7. Buy gomb – külső oldalra navigálás	4
5. A konfigurációs oldal (revuelto.html)	5
5.1. Háttérvídeo	5
5.2. A rendelési űrlap felépítése	5
5.3. JSON-ból modellbetöltés	5
5.4. jQuery alapú validáció és összefoglaló	6
5.5. Hibajelzés és vizuális visszajelzés	6
6. A Revuelto.json fájl példája	6
7. Külső könyvtárak	7
8. Összegzés	7

1. Bevezetés

A modern weboldalak napjainkban már messze túlmutatnak a statikus tartalom megjelenítésén. A HTML5, CSS3, JavaScript és külső könyvtárak (Swiper.js, Font Awesome) kombinációjával látványos, interaktív, mobilbarát prezentációk készíthetők. A jelen feladat egy luxus sportautókat bemutató egyoldalas webalkalmazás, amely Swiper.js alapú carousel-t, animációkat, overlay videó lejátszót, dinamikus címváltoztatást és részletes műszaki adatlapot tartalmaz.

2. Feladat leírása

A dokumentum egy Lamborghini, McLaren és BMW sportautókat bemutató, látványos egyoldalas weboldalt ír le, amely HTML, CSS, JavaScript és a Swiper.js könyvtár segítségével készült. A cél egy prémium autóipari stílusú, teljesen reszponzív, sötét témajú prezentáció létrehozása, ahol a felhasználó böngészheti az autókat, megnézheti a videókat, és konfigurálásra továbbmehet.

3. Weblap felépítése

3.1. Fejléc és főcím

A fejléc fixen a tetején helyezkedik el, tartalmazza a hamburger ikont és nyelvválasztót. A főcím dinamikusan változik az aktív slide alapján (JavaScript `changeTitle()` függvény).

3.2. Swiper carousel

A Swiper.js két példánya fut: egy fő carousel (`.bannerSwiper`) és egy thumbnail sáv (`.thumbsSwiper`), amelyek szinkronban vannak.

3.3. Interaktív elemek

- EXPLORE gomb – részletes nézet megnyitása
- Külső gomb – műszaki adatok megjelenítése
- Buy gomb – külső konfigurátor oldalra navigálás
- Lejátszás gomb – videó overlay megjelenítése

4. Scriptek

4.1. Autók adatai objektumban

```

1 var nevek = {
2     0: { cim: "KONFIGURALD <br>REVUELTO", alcim: "FROM NOW ON", ... },
3     1: { cim: "KONFIGURALD <br>HURACAN STO", ... },
4     ...
5 };

```

Az objektum tárolja az egyes autók címét, alcímét, videó URL-jét és logóját.

4.2. Swiper inicializálása

```

1 var thumbsSwiper = new Swiper(".thumbsSwiper", { ... });
2 const swiper = new Swiper(".bannerSwiper", {
3     thumbs: { swiper: thumbsSwiper },
4     speed: 1000,
5     ...
6 });

```

A thumbnail és a fő carousel összekapcsolása a thumbs opcióval.

4.3. Dinamikus cím és logó váltás

```

1 changeTitle = (index) => {
2     const data = nevek[index];
3     document.querySelector("#title").innerHTML = '<h1>${data.cim}</
4         h1>';
5     document.querySelector("#subtitle").innerHTML = '<p>${data.
6         alcim}</p>';
7     document.querySelectorAll('img.logo').forEach(img =>
8         img.src = 'kepek/${data.logo}');
9 };

```

Listing 1. Dinamikus cím és logó váltás

4.4. EXPLORE gomb működése

Az EXPLORE gomb elrejti a thumbnail sávot és a főoldali címeket, hogy helyet adjon a részletes slide-nak.

```

1 explore.forEach((el) => {
2     el.addEventListener("click", function() {
3         thumbs.classList.add("hide");
4         contents.forEach((e) => e.classList.add("hide"));
5         headers.forEach((e) => e.classList.add("hide"));
6         el.classList.add("hide");
7     });
8 });

```

4.5. Műszaki adatok megjelenítése

A Külső gomb a details osztály hozzáadásával nyitja meg az adatlapot, és megjeleníti a Buy gombot.

```

1 slideBtns.forEach(el => {
2     el.addEventListener("click", function() {
3         document.querySelector(".swiper-slide-active")
4             .classList.add("details");
5         let buyBtn = activeSlide.querySelector(".buy-btn");
6         if (buyBtn) buyBtn.style.display = "block";
7     });
8 });

```

4.6. Videó overlay kezelése

```

1 var showTrailer =() => {
2     var index = swiper.activeIndex;
3     videoContainer.innerHTML =
4         '<video controls autoplay>
5             <source src="kepek/${nevek[index].videoURL}" type="video/mp4">
6         </video>';
7     overlay.classList.add("show");
8 };
9
10 var closeOverlay =() => {
11     document.getElementById("video").pause();
12     overlay.classList.remove("show");
13 };

```

Listing 2. Videó overlay megnyitása és bezárása

4.7. Buy gomb – külső oldalra navigálás

```

1 var externalPages = ["revuelto.html", "sto.html", ...];
2
3 buyBtns.forEach((btn, index) => {
4     btn.addEventListener("click", function() {
5         window.location.href = externalPages[index];
6     });
7 });

```

Listing 3. Külső konfigurátor oldalakra ugrás

5. A konfigurációs oldal (revuelto.html)

5.1. Háttérvideo

```

1 <video autoplay muted loop id="background-video">
2   <source src="kepek/revuelto.mp4" type="video/mp4">
3 </video>
```

Autoplay, nemított, végtelenített háttérvideo adja a prémium hatást.

5.2. A rendelési űrlap felépítése

Az űrlap tartalmazza:

- Személyes adatok (név, email, cím)
- Modellválasztás (JSON-ból dinamikusan töltve)
- Színválasztó (<input type="color">)
- Dátumválasztó
- Extrák (checkbox)
- Fizetési mód (radio)

5.3. JSON-ból modellbetöltés

```

1 $.ajax({
2   url: "ajax/Revuelto.json",
3   dataType: "json",
4   success: function(data) {
5     data.cars.forEach(function(car) {
6       $("#model").append(
7         '<option value="${car.name}">${car.name}</option>' +
8       );
9     });
10   });
11 });
```

Listing 5. Modellválaszték betöltése JSON-ból

5.4. jQuery alapú validáció és összefoglaló

A beküldéskor client-oldali validáció fut:

```

1  $("#carOrderForm").submit(function(e) {
2      e.preventDefault();
3      let valid = true;
4
5
6
7      if(valid) {
8          const selectedCar = carData.find(car => car.name === $("#" +
9              "model").val());
10         const price = selectedCar ? selectedCar.price : "Ismeretlen
11             ";
12
13         const extras = $("input[name='extras']:checked")
14             .map(function(){ return this.value; }).get().join(", ")
15             || "Nincs";
16
17         const summary = '<h3>Rendeles osszefoglalo:</h3>
18             <p><strong>Ar: ${price} Ft</strong></p>
19             <p>Szin: <span style="color:${$("#color").
20                 val()}">
21                 ${$("#color").val()}</span></p>
22             <p>Extrak: ${extras}</p>';
23
24         $("#order-summary").html(summary).fadeIn();
25     }
26 });

```

Listing 6. Úrlap validáció és rendelés-összefoglaló

5.5. Hibajelzés és vizuális visszajelzés

Hibás mezőknél piros keret és egyedi hibaüzenet jelenik meg:

```

1 .error { border: 2px solid red !important; }
2 .error-message { color: red; font-size: 0.9em; }

```

6. A Revuelto.json fájl példája

```

1  {
2      "cars": [
3          {
4              "name": "Revuelto Standard",
5              "price": "189000000"
6          },
7          {
8              "name": "Revuelto S",
9              "price": "215000000"
10         },
11         {
12             "name": "Revuelto Performante",
13             "price": "248000000"
14         }
15     ]
16 }
```

1. Listing. Példa JSON struktúra

Listing 7. Revuelto.json fájl

7. Külső könyvtárak

- **Swiper.js v10** – carousel és thumbnail kezelés
- **Font Awesome 7** – ikonok
- **Google Fonts – Lato** – betűtípus

8. Összegzés

A weboldal teljes mértékben kihasználja a modern webtechnológiák nyújtotta lehetőségeket:

- Dinamikus tartalom JavaScript objektumból
- Szinkronizált carousel és thumbnail
- Animált átmenetek CSS-ben és JavaScriptben
- Videó overlay autoplay-jal
- Külső konfigurátor oldalakra mutató linkek
- Teljesen reszponzív, prémium megjelenés

A kód tiszta, jól strukturált, könnyen bővíthető további autómodellekkel.