

1) Kurzus-katalógus (alap DOM, szűrés és rendezés belső tömbből)

Cél

Készíts egy kétoldalas mini alkalmazást, amely megjelenít néhány egyetemi kurzust, kereshetően és rendezhetően, kizárólag a böngésző memóriájában lévő tömbből.

Fájlstruktúra

- kurzus-katalogus/
 - index.html
 - uj-kurzus.html
 - styles.css
 - app.js

Követelmények

index.html

- Fejléc: „Kurzus-katalógus” címmel és menüvel („Kezdőlap” és „Új kurzus”).
- Táblázat: a belső JS tömb kurzusait jeleníti meg.
- Oszlopok: Kód, Név, Kar, Kredit, Oktató.
- Keresőmező (<input type="search">): a JS if/else feltétellel szűr.
- Táblázat fejléce kattintható:
 - a JS switch szerkezet alapján rendez string szerint.
 - (pl. sortKey = "nev" | "kod" | "kredit" stb.)
- Ha nincs találat, egy rejtett <p id="msg"> jelenjen meg.

uj-kurzus.html

- Űrlap, amely bekéri a kurzus adatait.
- A beküldés (onsubmit) csak alert() üzenetet ad: nincs mentés.
- Validáció if/else alapon – üres mezők piros háttér.

styles.css

- Sötét téma, rácsos táblázat-stílus.
- .active menüelem jelölése.
- .card, .table-wrap, .muted stílusok.

app.js

- Belső tömb (kurzusok[]) 3–5 objektummal.

- for ciklussal építi fel a DOM-ot.
- if/else a kereséshez, switch a rendezéshez.
- Függvények:
 - render() – frissíti a táblázatot,
 - validateCourse() – ellenőrzi az űrlapot.

2) Éttermi menü és rendelés (kosár csak memóriában)

Cél

Egy kétoldalas webalkalmazás, ahol a menü táblázatából kosárba tehetőek az ételek.

A kosár csak futásidőben létezik (tömb a memóriában).

Fájlstruktúra

- etterem/
 - index.html
 - kapcsolat.html
 - styles.css
 - cart.js

Követelmények

index.html

- Táblázat ételekkel és árakkal.
- Minden sorban egy „Kosárba” gomb (<button class="add">).
- Jobb oldali panel (aside):
 - felsorolja a kosár tartalmát (<ul id="cart">),
 - mutatja az összesítést (<strong id="sum">),
 - link a „Rendelés leadása” oldalra.

kapcsolat.html

- „Asztalfoglalás” űrlap: név, e-mail, dátum, fő.
- Ellenőrzés validateContact() függvénnyel (if/else + alert).
- Jobb oldali oszlopban a kosár tartalma (memóriából újraépül a futásban).

cart.js

- Tömb cart[] a rendeléshez.

- Függvények:
 - addToCart(name, price)
 - removeAt(i)
 - sum() – for-ciklussal számol összeget
 - renderCart() – frissíti a DOM-ot
 - validateContact()
- Nincs localStorage vagy JSON.

styles.css

- Grid (2 oszlop) elrendezés.
- .card, .btn-secondary, .highlight osztályok.
- Reszponzív töréspont @media (max-width:840px).

3) Utazási iroda – táblázat + átadás query stringgel

Cél

Az index-oldal táblázatban listázza a belföldi utak célállomásait.

Sor-kattintás után átirányít foglalas.html?city=Eger formátumban.

Fájlstruktúra

- utazas/
 - index.html
 - foglalas.html
 - styles.css
 - app.js

Követelmények

- **index.html:**
 - JS tömb: {city, days, price} elemek.
 - for-ral táblázat generálása.
 - onclick esemény: location.href = "foglalas.html?city="+city.
- **foglalas.html:**
 - Az URLSearchParams segítségével kiolvassa a city paramétert.
 - Kiírja a választott várost (document.getElementById("city").innerText = ...).

- Egyszerű űrlap: név, dátum, személyek.
- validateForm() függvény if/else validálással.
- **Nincs tárolás.**
- CSS: egységes dizájn, két szakasz (információ + űrlap).

4) Könyvtár – keresés + „paginator” számolással (memóriában)

Cél

Egy könyvlista kereshető és lapozható, mindent JS-tömbből építünk.

Fájlstruktúra

- könyvtár/
 - index.html
 - ujkonyv.html
 - styles.css
 - app.js

Követelmények

- **index.html**
 - Keresőmező (cím vagy szerző).
 - Lapozó gombok („Előző”, „Következő”).
 - Egyszerre 5 sor látszik.
 - A for-ciklus számolja a kezdő-záró indexet (start = page*5, end = start+5).
- **ujkonyv.html**
 - Egyszerű űrlap (cím, szerző, év).
 - validateForm() – if/else-szel ellenőriz (üres mezők piros).
 - Csak alert, nincs mentés.
- **app.js**
 - renderPage() függvény: újrarajzolja az adott 5 rekordot.
 - Keresés filter helyett if/else logikával.
- **CSS**
 - Központi táblázat, .active-page kiemelés.

5) Edzőterem órarend – kiemelés (futásidőben) + jelentkezési űrlap

Cél

- Az index-oldal táblázatos órarendet mutat.
- A cellákra kattintva az adott óra kiemelhető (.fav toggle).
- A kiválasztás csak memóriában marad.

Fájlstruktúra

- edzoterem/
 - index.html
 - jelentkezes.html
 - styles.css
 - app.js

Követelmények

- **index.html**
 - Táblázat nap/oszlop szerkezetben.
 - onclick esemény: `classList.toggle("fav")`.
 - A kiválasztott órák neveit `selected[]` tömbbe tárolja.
 - Link a jelentkezési oldalra.
- **jelentkezes.html**
 - Megjeleníti a korábban (ugyanabban a futásban) kiválasztott órák listáját.
 - for-ciklussal építi fel.
 - Egyszerű űrlap (név, dátum, megjegyzés).
 - Validálás if/else.
- **Megjegyzés:** nincs tárolás vagy adatátvitel – cél a DOM és logikai gyakorlás.
- **CSS**
 - .fav osztály: eltérő háttérszín.
 - .card megjelenés, táblázat-elrendezés.

6) Mozi műsor – ülésválasztó (switch + ternary + ciklus)

Cél

Kétoldalas webalkalmazás:

- az első oldal a filmek listáját mutatja,
- a második egy „ülésválasztó” rendszer (JS-logika gyakorlása).

Fájlstruktúra

- mozi/
 - index.html
 - foglalas.html
 - styles.css
 - app.js

Követelmények

index.html

- Táblázat filmekkel (Cím, Időpont, Terem, Ár).
- Sorra kattintás → foglalas.html?film=....

foglalas.html

- URLSearchParams segítségével megjeleníti a választott filmet.
- Ülérács: pl. 5 sor × 8 szék, <button> elemekből.
- Kattintásra:
 - a gomb színe vált (.selected toggle),
 - a kiválasztott székek számát for-ciklussal összesíti.
- A kiválasztott szám alapján switch-ben dönt:
 - 0 → „Nem választottál szék.”
 - 1 → „1 székét választottál.”
 - több → „Több szék kiválasztva.”
- Egyszerű űrlap (név, e-mail) validációval (if/else).

styles.css

- .seat és .selected stílusok.
- Rácsszerű elrendezés (display: grid; grid-template-columns: repeat(8, 1fr)).
- Középre igazított tartalom.