

JEGYZŐKÖNYV

WEB TECHNOLÓGIA ALAPJAI

A fénykép(ező) titkai

Készítette: **Deme Zalán**

Neptunkód: **NDZDH6**

Dátum: 2024. május 05.

Miskolc, 2024

Tartalomjegyzék

Bevezetés	- 3 -
A feladat leírása.....	- 3 -
1. feladat – rendszerezés.....	- 4 -
1.1 – Fájlok rendszere	- 4 -
1.2 – Oldalak elrendezése.....	- 5 -
2. feladat – megvalósított feladatok.....	- 7 -
2.1 – CSS.....	- 7 -
2.2 – jQuery animáció	- 8 -
2.3 – JSON fájl.....	- 8 -
2.4 – AJAX, új HTML elem.....	- 9 -
2.5 – Egyedi videókezelőfelület	- 9 -
2.6 – Űrlap.....	- 10 -
2.7 – NodeJS konfigurálása.....	- 10 -

Bevezetés

A feladat leírása

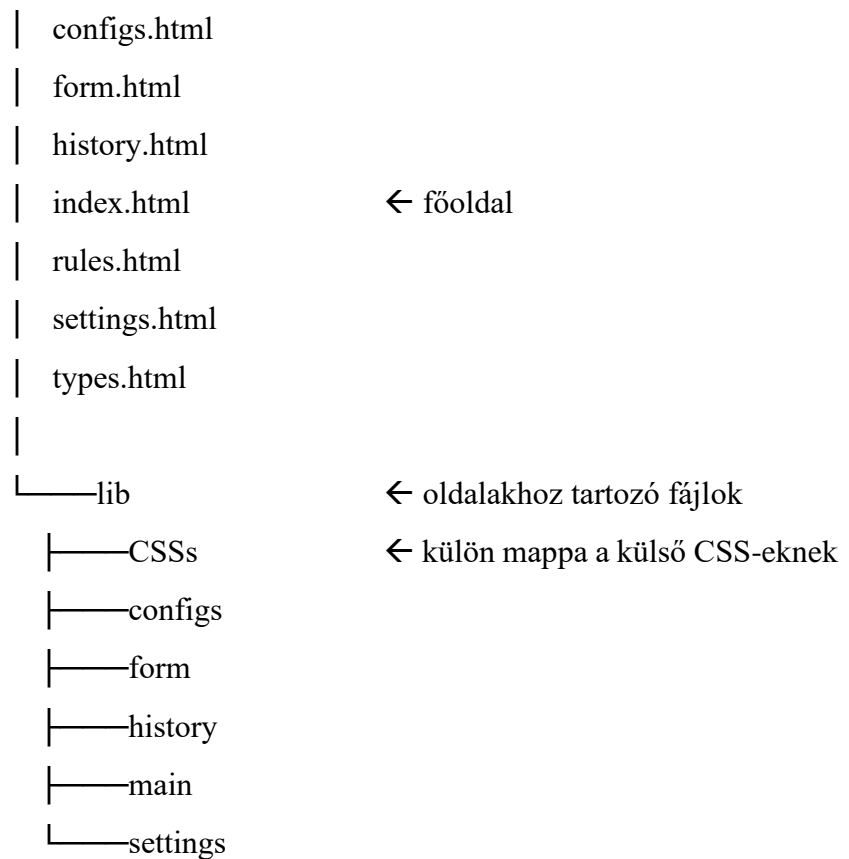
A feladatban egy több, legalább 5 html fájlból álló komplex weboldalt kellett elkészíteni előre egyeztetett témában. Az egyes oldalakat formázni kellett külső és belső CSS segítségével, videót és azt kezelő felületet kellett készíteni, továbbá egy űrlapot csinálni, aminek bemenő adatokat ellenőrzötten kellett beolvasni. Ezen feladatok megoldásához szükséges volt több JavaScript kód elkészítése is, melyek elengedhetetlenek voltak az oldal elvárt működéséhez és funkcióihoz. A weboldalakban továbbá még el kellett helyezni formázott táblázato(ka)t, jQuery animáció(ka)t, valamint egy adatokat tartalmazó JSON fájlt AJAX módszer segítségével megjeleníteni. Plusz feladatként lehetőség volt NodeJS konfigurálására.

A beadott feladatomban egy fényképezésről és fényképezőgépekről, kezdők számára szóló ismertető weboldalt készítettem el, melyben igyekszem egy átfogó képet adni az olvasóknak a fényképezés világáról. Az elkészült projekt 7 jól formázott weblapot tartalmaz, melynek mindegyike az egyik megadott feladatkritériumot teljesíti. Az elkészítés során nagy figyelmet fektettem arra, hogy jól átlátható, vonzó megjelenésű, és ránézésre is igényes weboldalt készítsek, hogy bárki, aki használja, könnyen tudjon benne tájékozódni, és közben meg is legyen elégedve az oldal kinézetével.

1. feladat – rendszerezés

1.1 – Fájlok rendszere

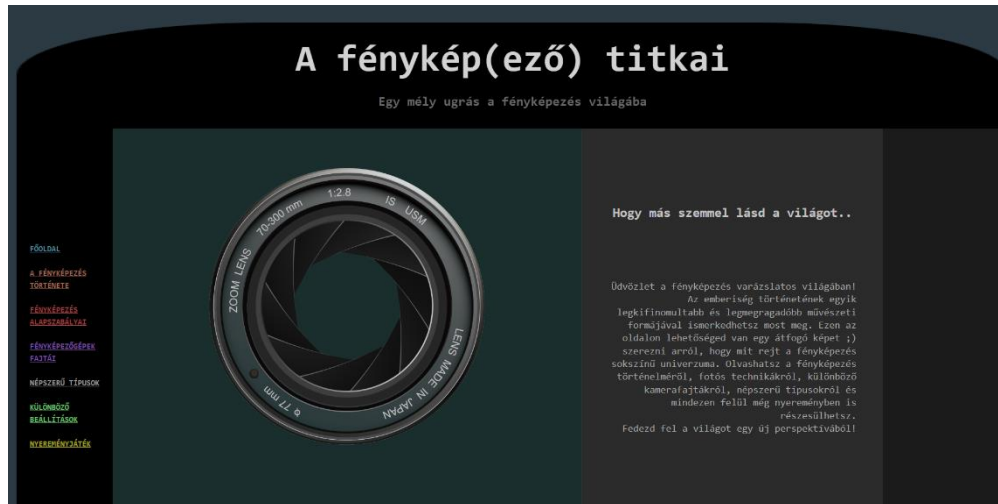
A weblaphoz szükséges fájlok (külső CSS-ek, külső JavaScript kódok, JSON fájl, képek, videók) mappákban elkülönítve, egy logikus mappastruktúrában vannak eltárolva, mely a következőképpen néz ki:



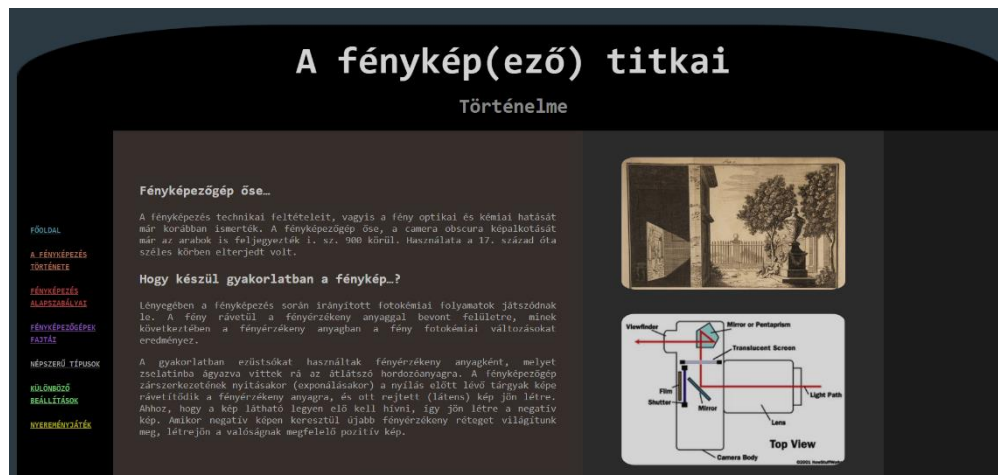
1.2 – Oldalak elrendezése

A weboldal a következő 5 működő oldalból épül fel:

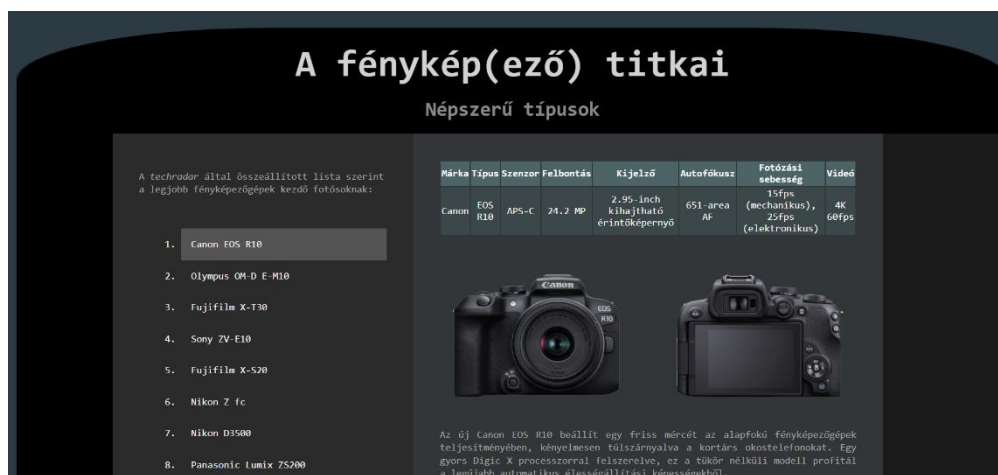
- **Főoldal**, ami tartalmaz egy rövid bemutatást, valamint leírja az egyes aloldalak célját, és lehetőséget ad a gyors eljutásra egy kattintással



- **A fényképezés története**, ahol a művészeti ág kialakulásáról lehet olvasni



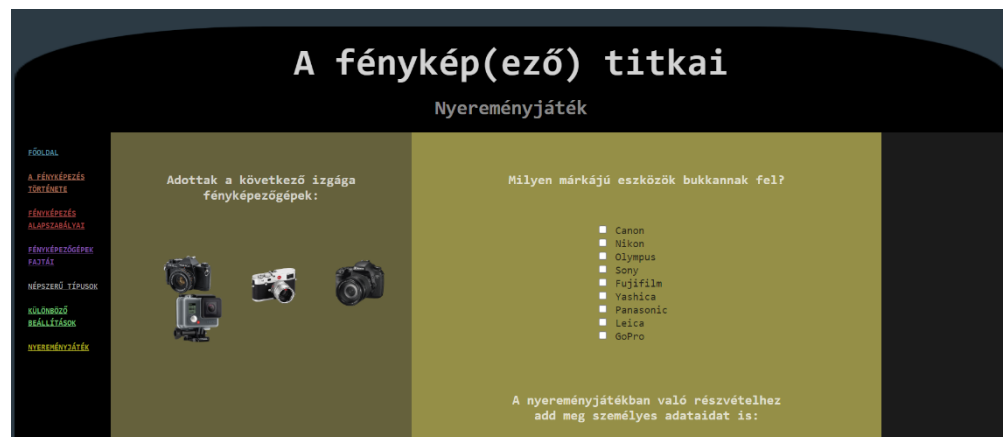
- **Népszerű típusok**, ahol kezdők számára ajánlott fényképezővázak vannak felsorolva



- **Különböző beállítások**, ahol szemléltetve van, hogy milyen beállítás milyen hatással lesz egy fényképre



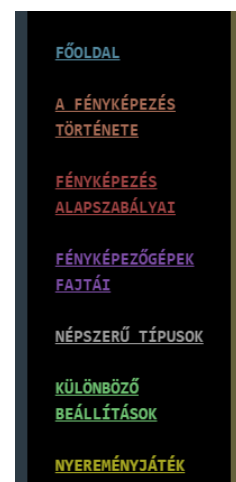
- **Nyereményjáték**, ahol űrlap helyes kitöltésével (az oldal szerint) lehet nevezni egy nyereményjátékra



Ezek az aloldalak az összes weboldalra jellemző elemek mellett (fej- és lábléc, navigációs sáv), mind egyénileg, az aloldal témájához igazítva lettek elkészítve és stílusozva.

Közös elemek:

- fejléc → projekt és az adott aloldal címe
- lábléc → források linkekkel, és a készítő (jómagam) neve
- navigációs sáv → innen bármelyik másik oldalra el lehet jutni, mindig látható, rögzített a pozíciója
- szélső sáv → az oldal alján rákattintva van lehetőség felugrani az oldal tetejére



2. feladat – megvalósított feladatok

Az egyes aloldalak általában egy-egy szükséges feladatot teljesítenek, némi átfedéssel

2.1 – CSS

Mindegyik oldal rendelkezik egy saját külső CSS fájljal, amik a többször előforduló (class alapú kiválasztás) és az egyszer előforduló (id alapú kiválasztás) elemek formázását tartalmazza, ezzel nagyban átláthatóbbá téve az egyes HTML kódokat, melyek így csak egy-kettő belső(inline) formázást tartalmaznak.

# configs.css	Az egyes aloldalak CSS fájljai
# form.css	
# history.css	
# main.css	
# rules.css	
# settings.css	
# types.css	

A main.css kódrészlete, alul példa az ID alapú kiválasztásra

```
html {
  scroll-behavior: smooth;
}

body {
  background-color: rgb(44, 58, 68);
}

header {
  background-image: url(../header.png);
  background-repeat: no-repeat;
  background-attachment: scroll;
  background-size: cover;
  background-size: 100% 100%;
  height: 120px;
  padding: 20px;
  text-align: center;
}

nav {
  float: left;
  width: 7%;
  height: 1380px;
  background: rgb(0, 0, 0);
  padding: 20px;
  text-align: left;
}

aside {
  float: right;
  width: 10%;
  height: 1380px;
  background: rgb(27, 27, 27);
  padding: 20px;
  text-align: center;
  font-size: 15px;
}

article#upperLeft {
  padding: 40px;
  width: 42%;
  background-color: #1a2e2d;
  height: 500px;
  text-align: center;
  float: left;
}
```

Példa inline CSS-re

```

```

2.2 – jQuery animáció

A **Főoldalon** egyből látható egy animáció, ahol a lencsében a rekeszek 360°-os fordulatot vesznek. Ez az animáció jQuery és egy paraméteres JavaScript függvény segítségével lett megoldva, ami önmagát hívja meg minden lefutáskor, így a rekesz a végtelenségig forog.

```
function rotateImage(degree) {  
    $('#rotatingImage').animate({ transform: degree }, {  
        step: function (now, fx) {  
            $(this).css({  
                '-webkit-transform': 'rotate(' + now + 'deg)',  
                '-moz-transform': 'rotate(' + now + 'deg)',  
                'transform': 'rotate(' + now + 'deg)'  
            });  
        },  
        duration:4000  
    }, 'linear');  
    setTimeout(() => rotateImage(360), 4000);  
} rotateImage(360);
```



2.3 – JSON fájl

A **Népszerű típusok** oldalon az adatok megjelenítéséhez egy külső JSON fájlból tölti be az oldal az egyes típusok adatai. A bodies.json fájl szerkezete:

```
{  
  "bodies": {  
    "body": [  
      {  
        "id": 1,  
        "brand": "Canon",  
        "type": "EOS R10",  
        "sensor": "APS-C",  
        "resolution": "24.2 MP",  
        "monitor": "2.95-inch kihajtható érintőképernyő",  
        "autofocus": "651-area AF",  
        "shootingRate": "15fps (mechanikus), 25fps (elektronikus)",  
        "video": "4K 60fps"  
      },  
      {  
        "id": 2,  
        "brand": "Olympus"
```


2.4 – AJAX, új HTML elem

Az előző weboldalon a JSON fájlt az oldal AJAX segítségével tölti be a weboldal, majd ezután új sort és elemeket fűz a táblázathoz.

```
function addRowToTable(bodyId) {
  const table = document.getElementById('cameras');

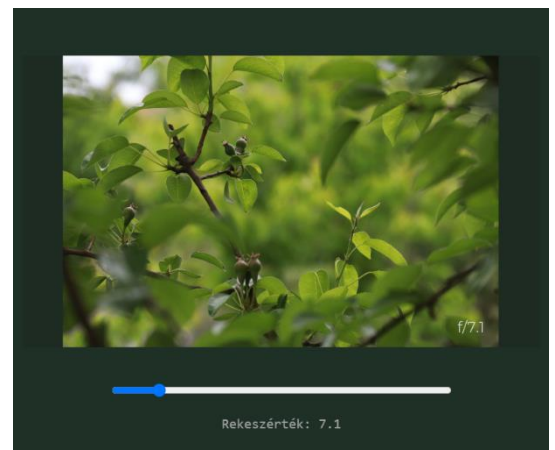
  $.getJSON("../lib/configs/bodies.json", function( jsonData ) {
    const cameraBody = jsonData.bodies.body.find(body => body.id === bodyId);

    const newRow = table.insertRow();
    newRow.innerHTML = `
      <td>${cameraBody.brand}</td>
      <td>${cameraBody.type}</td>
      <td>${cameraBody.sensor}</td>
      <td>${cameraBody.resolution}</td>
      <td>${cameraBody.monitor}</td>
      <td>${cameraBody.autofocus}</td>
      <td>${cameraBody.shootingRate}</td>
      <td>${cameraBody.video}</td>
    `;
  });
}
```

Márka	Típus	Szenzor	Felbontás	Kijelző	Autofókusz	Fotózási sebesség	Videó
Canon	EOS R10	APS-C	24.2 MP	2.95-inch kihajtható érintőképernyő	651-area AF	15fps (mechanikus), 25fps (elektronikus)	4K 60fps

2.5 – Egyedi videókezelőfelület

A **Különböző beállítások** aloldalon 3 videó szemlélteti a beállítások következményét, melyhez egyedi időszávot készítettem slider segítségével, hogy mégjobban meg lehessen nézni a változásokat, mivel ahogy ugrik a slider a következő pontra, oda ugrik a videó is. Emellett van lehetőség a videót teljes hosszában is megnézni. Felülírom az alapértelmezett input és output értékeit a slidernek, hogy a videóban ahhoz a másodperchez ugorjon, ami kell, valamint hogy az adott rekeszértéket írja ki, ne pedig a másodpercet.

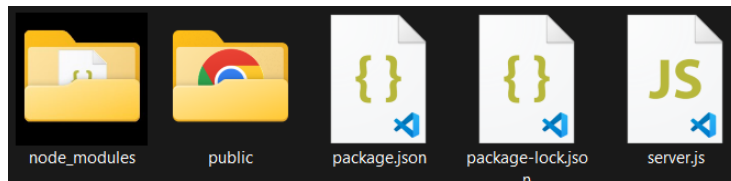


```
<input id="apertureInput" type="range" min="0" value="2" max="16" step="1"
  class="hideAperture"><br><br>
<div id="apertureOutput"
  style="color: ■ rgb(164, 164, 164); font-family: 'DDT Cond SemiBold', monospace;"
  class="hideAperture"></div>
```

```
const apertures = [5.6, 6.3, 7.1, 8, 9, 10, 11, 13, 14, 16, 18, 20, 22, 25, 29, 32, 36];
const apertureTimes = [1, 3, 6, 8, 11, 13, 16, 19, 21, 24, 26, 29, 31, 34, 36, 39, 41];
const apertureInput = document.getElementById('apertureInput');
const apertureOutput = document.getElementById('apertureOutput');

apertureInput.oninput = function () {
  apertureOutput.innerHTML = "Rekeszérték: " + apertures[this.value];
  apertureVideo.currentTime = apertureTimes[this.value];
};
apertureInput.oninput();
```


Ezután a fő mappában parancssorban kiadtam a *npm install* parancsot, amivel létrejött a *package-lock.json* fájl és a *node_modules* mappa és a beadandó fájljainak végleges rendszere:



Ezután a *npm start* paranccsal el lehetett indítani a szerveret, így böngészőben a <http://localhost:3000/> címen már el lehetett érni a weboldalt, a beérkezett válaszoknál pedig ki is íródtak a terminálra az adatok:

