

JEGYZŐKÖNYV

WEB TECHNOLÓGIA ALAPJAI

A fényképező titkai

Készítette: **Deme Zalán**

Neptunkód: **NDZDH6**

Dátum: 2024. május 10.

Miskolc, 2024

Tartalomjegyzék

Bevezetés	3
A feladat leírása	3
1. Struktúra	4
1.1 Fájlok rendszere	4
1.2 Oldalak elrendezése	5
2. Felhasznált elemek	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	10
2.6 Űrlap	11
2.7 NodeJS konfigurálása	12

Bevezetés

A feladat leírása

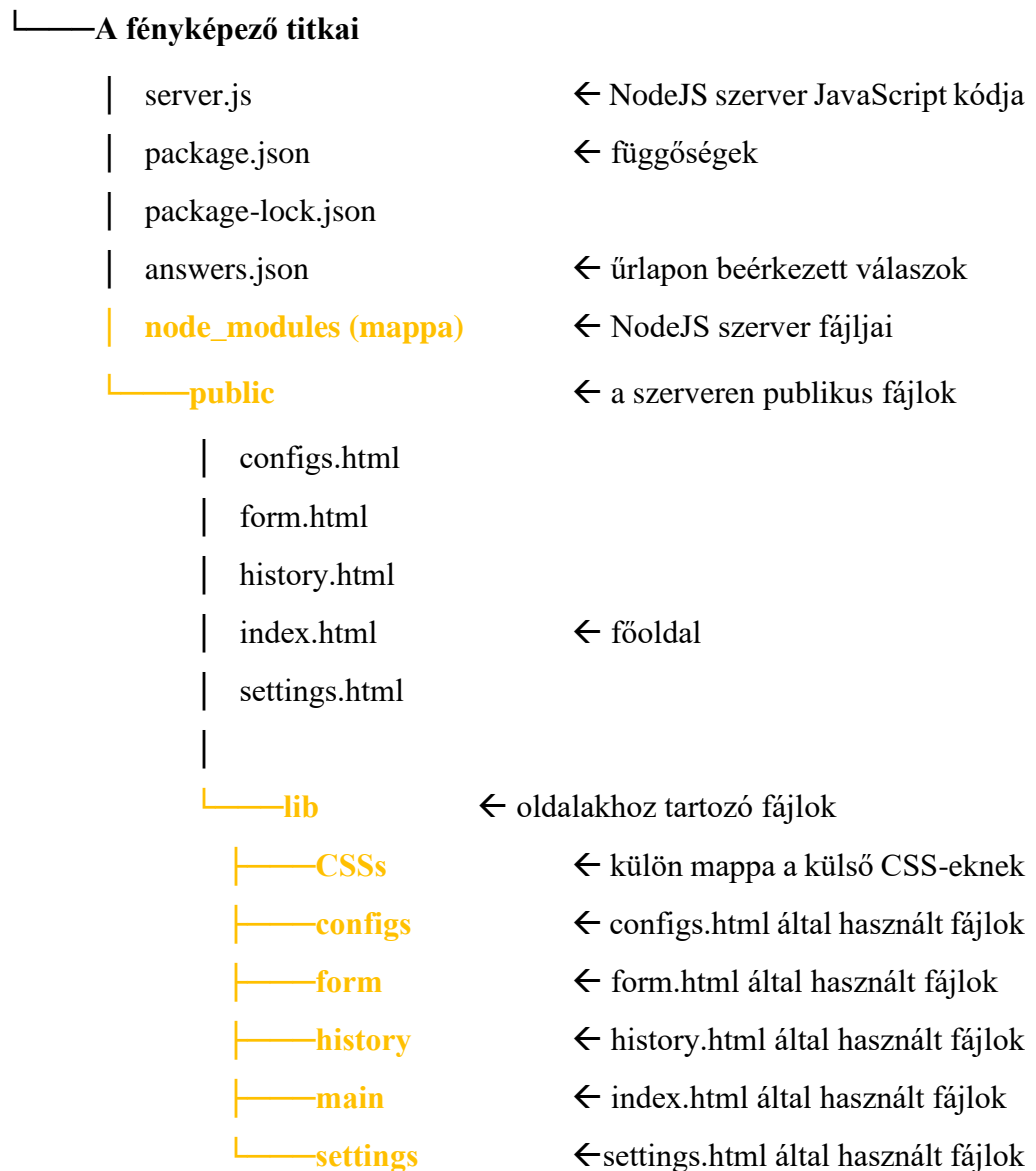
A beadott feladatomban egy fényképezésről és fényképezőgépekről, kezdők számára szóló ismertető weboldalt készítettem el, melyben lehet olvasni a fényképezés történelméről, népszerű típusokról, szemléltető videókat nézni, valamint egy játékos nyereményjátékot kitölteni, mindezzel igyekszem egy átfogó képet adni az olvasóknak a fényképezés világáról.

Az elkészült projekt 5 jól formázott, a html5 elemeit is használó html 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, amelyet külső és belső CSS segítségével valósítottam meg. A weboldalon található videókhoz továbbá készítettem egy egyedi videókezelő-felületet, valamint az aloldalon levő űrlaphoz egy ellenőrző metódust. Ezen funkciók megvalósításához elengedhetetlen volt JavaScript és jQuery kódok írása. Szorgalmi feladatként az elkészült weboldalak mellé konfiguráltam NodeJS szerveret is, amely segítségével a helyi 3000-es szerveren megjelenítettem a weboldalt.

1. Struktúra

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, segítve a könnyebb eligazodást, mely a következőképpen néz ki:



1.2 Oldalak elrendezése

A weboldal a következő 5 működő oldalból épül fel:

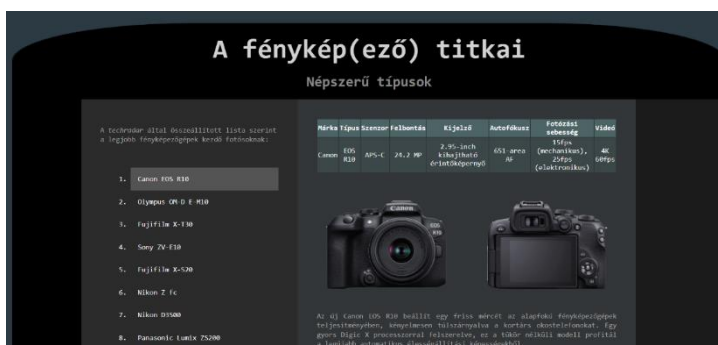
- **Főoldal**, ami tartalmaz egy rövid bemutatást, valamint leírja az egyes aloldalak célját, és lehetőséget ad a gyors eljutásra egy kattintással (1. ábra)
- **A fényképezés története**, ahol a művészeti ág kialakulásáról lehet olvasni (2. ábra)
- **Népszerű típusok**, ahol kezdők számára ajánlott fényképezővázak vannak felsorolva (3. ábra)
- **Különböző beállítások**, ahol szemléltetve van, hogy milyen beállítás milyen hatással lesz egy fényképre (4. ábra)
- **Nyereményjáték**, ahol űrlap helyes kitöltésével (az oldal szerint) lehet nevezni egy nyereményjátékra (5. ábra)



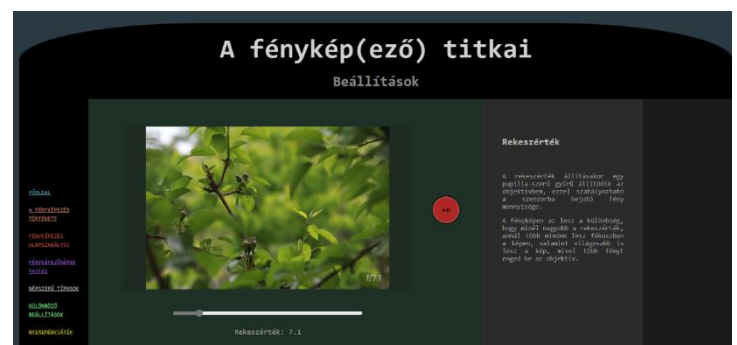
1. ábra



2. ábra



3. ábra



4. ábra

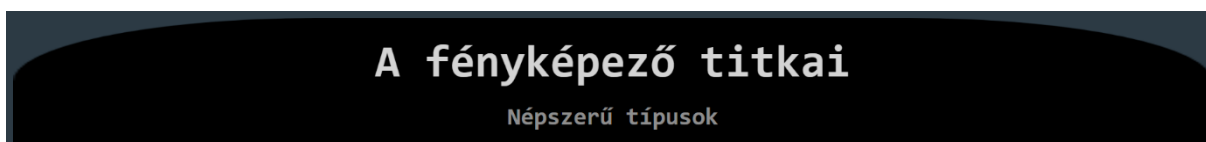


5. ábra

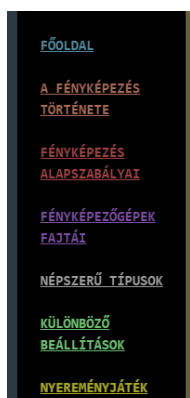
Ezek az oldalak az összes weboldalra jellemző elemeket leszámítva (fej- és lábléc, navigációs sáv), mind egyénileg, az oldal témájához igazítva lettek elkészítve és stílusozva.

Közös elemek:

- fejléc (header) → projekt és az adott oldal címe (6. ábra)
- navigációs sáv (nav) → innen bármelyik másik oldalra el lehet jutni, mindig látható, rögzített a pozíciója (7. ábra)
- tartalomrész (article) → az adott oldalhoz tartozó tartalom (8. ábra)
- szélső sáv (aside) → az oldal alján rákattintva van lehetőség felugrani az oldal tetejére (9. ábra)
- lábléc (footer) → források linkekkel, és a készítő (jómagam) neve (10. ábra)



6. ábra



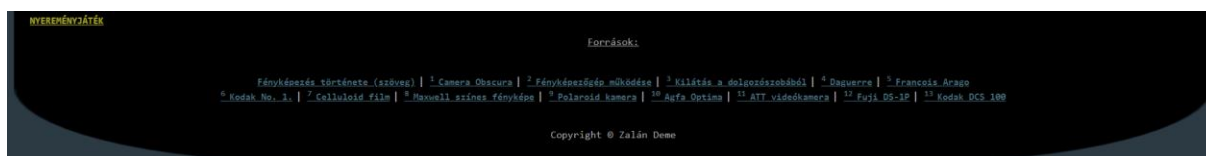
7. ábra



8. ábra



9. ábra



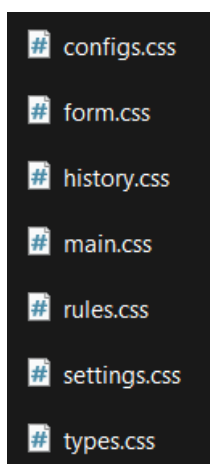
10. ábra

2. Felhasznált elemek

Az egyes aloldalak általában a beadandó egy-egy szükséges feladatát teljesítik, némi átfedéssel.

2.1 CSS

Mindegyik oldal rendelkezik egy saját külső CSS fájljal (11. ábra), amik a többször előforduló (class alapú kiválasztás) és az egyszer előforduló (id alapú kiválasztás, 12. ábra) elemek formázását tartalmazza, ezzel nagyban átláthatóbbá téve az egyes HTML kódokat, melyek így csak egy-kettő belső (inline) formázást (13. ábra) tartalmaznak.



11. ábra

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

body {
  background-color: #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: #0, 0, 0;
  padding: 20px;
  text-align: left;
}

aside {
  float: right;
  width: 10%;
  height: 1380px;
  background: #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;
}
```

12. ábra

```

```

13. ábra

2.2 jQuery animáció

A **Főoldalon** egyből látható egy animáció (14. ábra), ahol a lencsében a rekeszek 360°-os fordulatot vesznek. Ez az animáció jQuery és egy paraméteres JavaScript függvény segítségével lett megoldva (15. ábra), ami önmagát hívja meg minden lefutáskor, így a rekesz a végtelenségig forog.



14. ábra

```
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);
```

15. ábra

2.3 JSON fájl

A **Népszerű típusok** oldalon az adatok megjelenítéséhez egy külső JSON fájlból (16. ábra) tölti be az oldal az egyes típusok adatai. A **Nyereményjáték** oldalon az elküldött válaszokat a NodeJS szerver az answers.json fájlba tárolja el (17. ábra).

```
{
  "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"
      }
    ]
  }
}
```

16. ábra

```
{
  "camera": [
    "canon",
    "olympus",
    "fujifilm",
    "yashica"
  ],
  "name": "Alajos",
  "birthday": "1969-07-12",
  "gender": "male",
  "message": "Koppkopp..\\r\\n",
  "favcolor": "#48e577",
  "brand": "bfb"
}
```

17. ábra

2.4 AJAX, új HTML elem

A **Népszerű típusok** weboldalon a JSON fájlt az oldal AJAX segítségével tölti be a weboldal (18. ábra), majd ezután új sort és elemeket fűz a táblázathoz, így kerülnek megjelenítésre az oldalon az adatok (19. ábra).

```
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>
    `;
  });
}
```

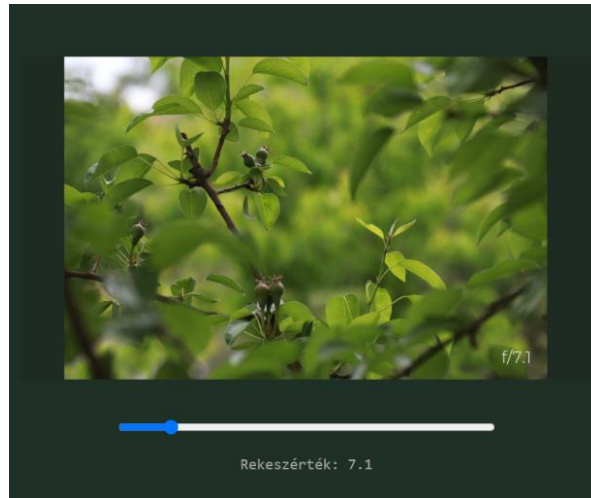
18. ábra

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

19. ábra

2.5 Egyedi videókezelőfelület

A **Különböző beállítások** aloldalon 3 videó (20. ábra) szemlélteti a beállítások következményét, melyhez egyedi időszávot készítettem slider (21. ábra) 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 (22. ábra).



20. ábra

```
<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>
```

21. ábra

```
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'),
      apertureOutput = document.getElementById('apertureOutput');

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

22. ábra

2.6 Ürlap

A **Nyereményjáték** aloldal egy űrlap segítségével imitálja a nyereményjáték érzését. Ez az űrlap tartalmazza azokat az ellenőrzéseket, amik egy nyereményjátékhoz is szükségesek. A név, dátum és a nem bejelölését beépített html tulajdonságok hozzáadásával oldottam meg (23. *ábra*), arra pedig, hogy 4 válasszal fogadja csak el az oldal a beküldést, egy JavaScript kódot írtam (24. *ábra*).

[illegible]

23. ábra

```
$(document).ready(function () {
    $('#submit').click(function() {
        checked = $("input[type=checkbox]:checked").length;

        if(checked != 4) {
            alert("4 márkát kell bejelölni!");
            return false;
        }
    });
});
```

24. ábra

2.7 NodeJS konfigurálása

Miután telepítettem a számítógépre, a projekt mapparendszerét kellően átstrukturáltam, hogy működhessen a NodeJS, így minden eddigi fájlt beleraktam egy public nevű mappába, emellé pedig megírtam a *server.js* (25. ábra) és a *package.json* fájlt (26. oldal).

A *server.js* kódban beállítom, hogy a public mappa a szerveren látható legyen, valamint hogy indításkor az *index.html* oldalt töltse be. Mindezt a helyi 3000-es porton jeleníti meg a szerver.

```
const express = require('express');
const path = require('path');

const app = express();
const PORT = process.env.PORT || 3000;

app.use(express.static(path.join(__dirname, 'public')));
app.use(express.urlencoded({ extended: true }));

app.get('/', (req, res) => {
  res.sendFile(path.join(__dirname, '/public/index.html'));
});
```

25. ábra

```
{
  "name": "fenykepezo_titkai",
  "version": "1.0.0",
  "description": "fenykepezokrol infok",
  "main": "server.js",
  "scripts": {
    "start": "node server.js"
  },
  "dependencies": {
    "express": "^4.17.1"
  }
}
```

26. ábra

Ha az oldal látogatója beküldi az űrlapon a válaszait, azok eltárolódnak egy JSON fájlba, ezt a NodeJS kezeli (27. ábra).

```
app.post("/form.html", (req, res) => {

  const fs = require('fs');
  const formData = req.body;
  const jsonString = JSON.stringify(formData, null, 2);

  fs.readFile('answers.json', 'utf8', (err, data) => {
    if (err) {
      console.error('Error reading file:', err);
      return;
    }

    let existingData = [];

    if (data.trim() !== '') {
      existingData = JSON.parse(data);
    }

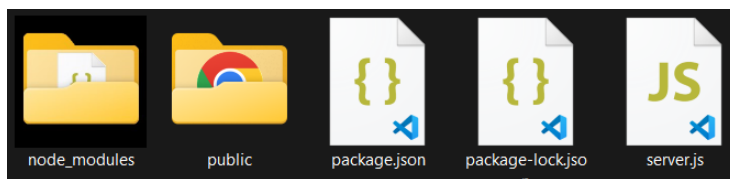
    existingData.push(formData);
    const updatedJsonString = JSON.stringify(existingData, null, 2);

    fs.writeFile('answers.json', updatedJsonString, (err) => {
      if (err) {
        console.error('Error writing file:', err);
      } else {
        console.log('A new submit has been recorded in answers.json');
      }
    });
  });
  res.redirect('/');
```

27. ábra

Ha sikeresen beküldtük válaszunkat, visszadob a szerver a főoldalra egy „Sikeres beküldés!” üzenet megjelenésével együtt.

Mindezek után 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 (28. ábra).



28. ábra

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 sikeresen el is tárolódtak azok az *answers.json* fájlban.