

JEGYZŐKÖNYV

Web technológia alapjai

Node.js 101, azaz egy gyors weboldal
készítése Express backendel

Készítette: **Kiss Konrád Soma**

Neptunkód: **CNY8MP**

Dátum: 2024. május 04.

Miskolc, 2024.

Tartalomjegyzék

| | |
|---|----------|
| Bevezetés | 3 |
| 1. Feladat: HTML | 4 |
| 1.1 Feladat: Megfelelő mennyiségű tartalom | 4 |
| 1.2 Feladat: Szerkezet megtervezése | 4 |
| 1.3 Feladat: Felhasználandó elemek | 4 |
| 1.4 Feladat: Űrlap | 4 |
| 1.5 Feladat: Videó | 4 |
| 2. Feladat: CSS | 5 |
| 2.1 – 2.3 Feladat: Megfelelő formázások alkalmazása | 5 |
| 2.4 Feladat: Táblázat formázása | 5 |
| 2.5 Feladat: Menü kialakítása | 5 |
| 2.6 – 2.8 Feladat: Háttérszín, linkek, űrlap elemek és gombok formázása | 5 |
| 2.9 Feladat: Külső CSS fájlban, és beágyazva is legyen CSS alkalmazva | 5 |
| 3. Feladat: JavaScript, jQuery, JSON és AJAX | 5 |
| 3.1 Feladat: Form ellenőrzés | 5 |
| 3.2 Feladat: jQuery animáció | 5 |
| 3.3 Feladat: elemek kiválasztása | 5 |
| 3.4 – 3.6 Feladat: új html elem készítése, meglévő módosítása, JSON adattárolás, AJAX alkalmazásával a JSON fájl megjelenítése az oldalon | 6 |
| 4. Feladat: Szorgalmi | 6 |

Bevezetés

A feladat leírása: A feladat egy HTML5-ön alapuló weboldal készítése volt, amely egy választott téma keretein belül eleget tesz az előre megszabott kritériumoknak.

Ezek a kritériumok a felhasznált eszközökről, az oldal elrendezéséről és minimum tartalmáról szólnak. A választott témából (Node.js 101) kifolyólag arra is oda kell figyelni, hogy a saját magunk által megszabott tartalmi szükségleteket is kielégítse az oldal. Ebben az esetben ezek az alábbi dolgok voltak: a HTML5 alapvető szerkezetének és használatának bemutatása, CSS használatának bemutatása és alkalmazása a weboldalon, JavaScript és jQuery példák bemutatása és használata, JSON fájl felhasználása AJAX segítségével, az oldal interaktivitásának biztosítása további jQuery kód segítségével, űrlapok használata és kezelése egy szorgalmi feladatként elkészített Node.js backend segítségével.

A kész projekt fájlrendszeri elrendezése az alábbi lett:

- A projekt gyökérkönyvtára: **WebTechCNY8MP**
 - A statikus fájlokat tároló mappa: **public**
 - Az interaktivitást biztosító fájlokat tároló mappa: **lib**
 - A **.html**, **.css** és **egyéb** tartalommal rendelkező fájl
 - A Node.js backend fő programfájlja: **index.js**
 - A felhasznált könyvtárakat beállító, és a projekt működését elősegítő **package.json** fájl

Minden más a mappában található fájl ideiglenes, vagy a működéshez elengedhető:

- **node_modules** mappa: a könyvtárakat tartalmazó mappa, rendkívül nagy és sok fájlból áll, ezért nem szokás a projekttel együtt feltölteni, hanem a **package.json** -ban feljegyezzük hogy mely könyvtárakat használtuk, és az **npm install** paranccsal letöltjük őket.
- **.gitignore**: a projekt feltöltéséhez szükséges fájl, az ideiglenes/nem feltöltendő mappákat és fájlokat szűri ki.
- **.github** mappa: a projekt esetleges külső szerveren történő futtatásához szükséges folyamatok leírását tartalmazza, lokális futtatás esetén szükségtelen.

1. Feladat: HTML

1.1 Feladat: Megfelelő mennyiségű tartalom

A feladat leírása azt kérte, hogy legalább öt HTML fájl legyen a projektben. A jegyzőkönyv írásakor a projekt az alábbiakat tartalmazza:

- index.html – főoldal
- html.html – HTML leírás
- css.html – CSS leírás
- js.html – JavaScript és jQuery leírás
- nodejs.html – Node.js leírás
- express.html – Express könyvtár leírás
- contact.html – Kapcsolatlétesítéshez készült űrlap
- about.html – A készítő elérhetőségei

1.2 Feladat: Szerkezet megtervezése

Az oldal szerkezete az alábbi:

- Egy felső címsor az oldal nevével.
- Egy bal oldali oldalsáv az elérhető oldalakkal.
- Egy jobb oldali oldalsáv egy szimulált bejelentkezés funkcióval.
- Egy középső tartalmi rész.
- Egy lábrész a készítő nevével és a készítés évével.

1.3 Feladat: Felhasználandó elemek

Az oldal az alábbi módon használja a szükséges elemeket:

- **div** – rengeteg részen, ahol előre kész elem nincs a feladatra, megfelelő elválasztásra használtam. Példa: a css.html oldalon a margó példát ezzel készítettem el.
- **span, p, címsorok** – az oldalak tartalma ezekkel az elemekkel van elrendezve, kiemelve.
- **képek**: a html.html-en, illetve minden oldalon a címsoron alkalmazva van egy szép megjelenés eléréséhez.
- **táblázat**: az about.html-en az adataim ebben vannak elrendezve.
- **linkek**: a cikkek során többször is alkalmaztam őket további hasznos információk megjelölésére.
- ...

1.4 Feladat: Űrlap

A szükséges űrlapelemeket mind alkalmaztam a contact.html oldalon.

1.5 Feladat: Videó

Egy beágyazott videó található a főoldalon, amely a témával kapcsolatos és elláttam egyedi irányításhoz használt gombokkal.

2. Feladat: CSS

2.1 – 2.3 Feladat: Megfelelő formázások alkalmazása

Attól függően, hogy milyen módon volt a legegyszerűbb megoldani, úgy alkalmaztam a megfelelő formázási kijelöléseket. Példák: a kódblokkok style attribútum segítségével vannak színezve, a főcím és pár elrendezési elem azonosító alapján van kijelölve, és pár ismétlődő elem osztály alapján.

2.4 Feladat: Táblázat formázása

Az about.html oldalon látható egy megformázott táblázat az adataimmal.

2.5 Feladat: Menü kialakítása

A bal oldali oldalsávon látható a menü, benne az összes elérhető oldallal.

2.6 – 2.8 Feladat: Háttérszín, linkek, űrlap elemek és gombok formázása

Mindezen formázások megtalálhatóak a **style.css** fájlban, és láthatóak az oldalon.

2.9 Feladat: Külső CSS fájlban, és beágyazva is legyen CSS alkalmazva

Szinte a projekt minden oldalán található példa a beágyazott CSS alkalmazására, és egy közös **style.css** fájljal lett elkészítve úgy az oldal, hogy végig konzisztens kinézete legyen.

3. Feladat: JavaScript, jQuery, JSON és AJAX

3.1 Feladat: Form ellenőrzés

A **contact.html** oldalon csak akkor engedi az oldal, hogy elküldjünk egy üzenetet, amennyiben minden helyen megfelelő adat található. Ha valahol hibás a bemenet, letiltásra kerül a küldés gomb, és egy üzenettel, illetve piros kerettel kiemeli a hibát az oldal.

Bár a HTML5 ad pár alapvető eszközt ezeknek az adatoknak a leellenőrzésére, egy kicsit robusztusabb, teljesen JavaScript alapú hibaellenőrzést készítettem.

3.2 Feladat: jQuery animáció

A jQuery segítségével három helyen készítettem apróbb animációkat. Az oldalsávok kinyílása, bezárása jQuery segítségével történik, illetve a **js.html** oldalon egy konkrét példát egy jobbra-balra animált elemmel mutatok az utasításokról.

3.3 Feladat: elemek kiválasztása

A feladat azt kérte, hogy a jQuery kijelölőket használjam HTML tag névvel, osztállyal és azonosítóval. Ezekre található példa a **kk.js** és **login.js** fájlban:

- HTML tag névvel kerülnek kijelölésre a **code** blokkok, ezzel kijelölve az összeset és alkalmazva rájuk a szintaxis színezést.
- Osztállyal és azonosítóval: a jobb oldali navigációs sáv gombját egy osztály és azonosító kombinációjával sikerül a legpontosabban kijelölni.

3.4 – 3.6 Feladat: új html elem készítése, meglévő módosítása, JSON adattárolás, AJAX alkalmazásával a JSON fájl megjelenítése az oldalon

Az összes oldalon megtalálható bal oldali oldalsáv mindhárom kritériumot kitölti az alábbi módon:

- Egy JSON fájlban található az összes hivatkozás, ezzel biztosítva, hogy megegyeznek az oldalakon keresztül.
- A meglévő, de még csak nyitó/csukó gombbal felszerelt oldalsávot **szerkesztem** az **újronnan létrehozott** navigációs gombok hozzáadásával.
- Mind ehhez a szükségét **content.json** fájl a jQuery AJAX metódusaival töltöm be.

4. Feladat: Szorgalmi

A szorgalmi feladat az volt, hogy az oldalt tegyük elérhetővé egy saját Node.js backend segítségével. Ezt az **index.js** fájlban belül értem el, amelyet a mappán belül lehet elindítani az **npm start** paranccsal.