

# JEGYZŐKÖNYV

Web technológiák 1

Kórház Időpontfoglaló

Készítette: **Sebe Zsolt**

Neptunkód: **ACC02G**

Dátum: 2025. december

**Miskolc, 2025**

## Tartalomjegyzék

1	Bevezetés .....	3
2	Jegyzékek .....	3
3	Használat .....	3
4	HTML Struktúra .....	4
4.1	Scriptek .....	4
5	API .....	5
6	CDN-barát felépítés .....	5
7	Összegzés .....	6

## 1 Bevezetés

A projekt célja egy egyszerű, webes kórházi időpontfoglaló rendszer bemutatása. A felhasználói felület kialakításához a TailwindCSS keretrendszerét és a jQuery könyvtárat használtuk, hogy gyorsan és rugalmasan lehessen létrehozni a dinamikus interakciókat, mint például az időpontok lekérése és az űrlapok validálása.

A projekt szemlélteti az alapvető front-end és back-end kommunikációt, JSON alapú API-k használatát, valamint egyszerű szerveroldali feldolgozást Node.js segítségével.

## 2 Jegyzékek

A projekt jegyzék struktúrája a következő:

- `public/`: A nyilvánosan elérhető fájlok mappája, amelyeket a webserver szolgál ki. Tartalmazza:
  - `scripts/`: JavaScript fájlok, például a front-end interakciók és AJAX kérések.
  - `imgs/`: Képek és videók, melyek a felhasználói felületen jelennek meg.
- `src/`: Forrásfájlok, amelyeket build-álni kell (pl. Tailwind CSS forrás).
- `server.js`: Egyszerű Node.js szerver, amely kezeli a JSON alapú API hívásokat, például az időpontok lekérését és foglalását. Ez a szerver csak példaként szolgál, hogy bemutassa a backend működését.
- `package.json`, `package-lock.json`: NPM konfigurációs és zárolási fájlok, amelyek a projekt függőségeit tartalmazzák.

## 3 Használat

A projekt építése és futtatása egyszerű parancsokkal történik:

A stílusok generálásához futtassuk a következőt a projekt gyökerében:

```
npm run build
```

Ez létrehozza a kész `public/style.css` fájlt.

A `tailwind.css` fájlban egyéni stílusok is vannak.

A Node.js szerver futtatása a következő parancssal történik:

```
node server.js
```

Ezzel elindul a webalkalmazás, és a front-end az API hívásokkal kommunikál a szerverrel.

## 4 HTML Struktúra

A webalkalmazás HTML oldalai egységes szerkezetet követnek:

- **Header:** a navigációs menüt tartalmazza, állandó a teljes oldalon.
- **Main content:** a fő tartalom helye, itt változik az oldal specifikus része.

A main content részét a `<!-- Main Content -->` komment jelöli, és minden oldal (Kezdőlap,

Rólunk, Időpontfoglalás, Orvosaink, Kapcsolat) ezt a részt cseréli ki a saját tartalmával.

- **Footer:** az állandó lábléc, jogi és kapcsolat információkkal, minden oldalon azonos.

Ez a felépítés biztosítja, hogy az oldal egységes dizájnt és konzisztens navigációt használjon, miközben a fő tartalom oldalanként eltér.

### 4.1 Scriptek

Minden oldal az alábbi közös scriptet használja:

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.7.1/jquery.min.js" integrity="sha512-v2CJ7UaYy4JwqLDIrZUI/4hqeoQie0mAZNXBeQyjo21dadnwR+8ZaIJVT8EE2iyI610V8e6M8PP2/4hpQINQ/g==" crossorigin="anonymous" referrerpolicy="no-referrer"></script>
```

Ezek együtt biztosítják, hogy a külső script:

1. Ne legyen manipulált (`integrity`)
2. Ne küldjön fölösleges hitelesítő adatot (`crossorigin`)
3. Védeni tudja a felhasználói adatokat (`referrerpolicy`)

Ezért használjuk a modern weboldalakon a CDN-ekről betöltött JavaScript fájloknál.

Ha egy oldalhoz egyedi JavaScript szükséges, azt a `/scripts/<name>.js` fájlból töltjük be, ahol `<name>` az oldal HTML fájlának neve. Például a `contact.html` esetén:

```
<script src="scripts/contact.js" defer></script>
```

A `defer` attribútumnak köszönhetően a script csak az oldal betöltődése után fut le, így a DOM már elérhető, és a script nem blokkolja az oldal renderelését.

## 5 API

A projekt Node.js szerverén az alábbi JSON alapú API végpontok érhetők el:

**/api/make-appointment** Új időpont foglalására szolgál. A kliens oldali űrlap JSON formátumban küldi a szükséges adatokat (`name`, `email`, `date`, `time`, `fav-color`).

**/api/query-date** Egy adott naphoz tartozó szabad időpontokat lehet lekérdezni. A kliens JSON formátumban küldi a kért `date`-et, a szerver pedig visszaadja a naphoz tartozó időintervallumok listáját, amelyben az `available` mező jelzi, hogy az adott időpont foglalható-e.

**/api/sendmsg** Ezzel lehet üzenetet (pl. kapcsolatfelvételi mailt) küldeni. A kliens JSON objektumban küldi a feladót (`name`, `reply-to`), a tárgyat (`subject`) és a tartalmat (`content`).

szerver visszaad egy `success: true/false` választ, valamint hiba esetén a `errmsg` mezőt.

## 6 CDN-barát felépítés

Az oldal statikus elemei ( minden, ami nem az `/api/` útvonalhoz tartozik) teljes mértékben CDN-en tárolhatók, így a tartalom gyorsan és megbízhatóan érhető el a felhasználók számára.

Az API végpontok mellett JSON sémák is rendelkezésre állnak, amelyek lehetővé teszik, hogy a CDN előszűrje a bejövő kéréseket (“API Shield”), ellenőrizve azok szerkezetét, mielőtt azok az origin szerverhez érkeznének. Ez jelentősen növeli a biztonságot és csökkenti a szerver terhelését.

A Node.js szerver a helytelen URL-eket automatikusan átirányítja:

- `valami.html` → `valami`
- `valami/` → `valami`

A trailing slash eltávolítása biztosítja a konzisztens URL-szerkezetet, ami megfelel a modern CDN-ek által nyújtott automatikus HTML-kezelésnek és átirányítási szabályoknak, ahogy azt a Cloudflare dokumentációja is bemutatja: <https://developers.cloudflare.com/workers/static-assets/routing/advanced/html-handling/#drop-trailing-slashes>.

## 7 Összegzés

A projekt célja egy egyszerű kórházi időpontfoglaló rendszer bemutatása volt.