A képen szöveg, Betűtípus, képernyőkép, Grafika látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

Web Programozás I. beadandó feladat

Készítők:

Kordics Kristóf

ZE27GW

Krecsmarik Zalán András

B1XTCZ

Mérnökinformatikus szak

2025

A projekt GitHub URL-je: <https://github.com/Kkordics/WebProgramozasHazi>

A weboldal URL-je: <http://zalankristof.nhely.hu/index.html>

**Tartalom**

[1. Alapbeállítások 3](#_Toc195558940)

[2. Táblázat(CRUD) 3](#_Toc195558941)

[3. HTML5 4](#_Toc195558942)

[Web Storage: 4](#_Toc195558943)

[Web Workers: 5](#_Toc195558944)

[Server-Sent events: 6](#_Toc195558945)

[Geolocation API: 7](#_Toc195558946)

[Drag and drop API: 9](#_Toc195558947)

[Canvas SVG: 9](#_Toc195558948)

[4. AJAX 12](#_Toc195558949)

[5. OOJS 13](#_Toc195558950)

[6. React 15](#_Toc195558951)

[7. Admin bejelentkezés 18](#_Toc195558952)

# Alapbeállítások

A fejlesztés során a **Visual Studio Code**-ot használtuk és a benne elérhető **Live Server** plugin-t. Elsőként létrehoztunk egy base.html file-ot, ami reprezentálja az oldal alap szerkezetét. Ez az oldal jelen van az online tárhelyen de a publikus weboldalon semmilyen módon nem elérhető, a későbbi fejleszthetőség érdekében.

Web hosting: <https://www.nethely.hu>

* Felhasználó név: [kkordics@gmail.com](mailto:kkordics@gmail.com)
* Jelszó: xyz (nem a tényleges neptun kód)

FTP elérése (ftp.nethely.hu):

* Felhasználó név: xyz
* Jelszó: xyz

1. Táblázat(CRUD)
2. HTML5

## Web Storage:

A Web Storage a **Asztalfoglalás** oldalon került implementálásra. A főbb funkció ami a mentést végzi az a **storeCurrecntState()**. Ami elmenti az input mezők és az székek értékét.

function storeCurrecntState(){

    if (typeof(Storage) !== "undefined"){

        //input mezők

        localStorage.setItem("a\_rend\_nev", document.getElementById("a\_rend\_nev").value);

        localStorage.setItem("a\_rend\_email", document.getElementById("a\_rend\_email").value);

        localStorage.setItem("a\_rend\_tel", document.getElementById("a\_rend\_tel").value);

        localStorage.setItem("a\_rend\_date", document.getElementById("a\_rend\_date").value);

        //Székek

        for(let i = 1;i <=48;i++){

            if(document.getElementById("szek"+i.toString()).querySelector("p").innerHTML != ""){

                localStorage.setItem("szek"+i.toString(), document.getElementById("szek"+i.toString()).querySelector("p").innerHTML);

                console.log("asd");

            }

        }

    }else{

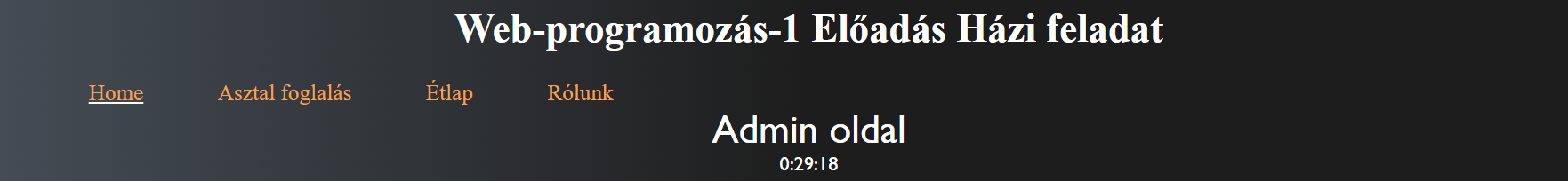
        console.error("A böngésző nem támogatja a local storage funkciót!");

    }

}

## Web Workers:

Az [admin](#_Admin_bejelentkezés) oldalon került implementálásra. Egy bejelentkezési visszaszámláló ként. Az admin felhasználónak 30 perce áll rendelkezésre, hogy az admin oldalon tevékenységeit végezze. A visszaszámláló kód a **countdown.js** file-ba található. Amint a munka menet lejárt a alert üzenetként egy figyelmeztető üzenet jelenik meg, amint ennek elfogadása után a **home** oldal jelenik meg.



Ami a script.js file-ba keröl meghívásra a következő módon:

var webWorker;

window.addEventListener("DOMContentLoaded", function () {

    if (window.location.pathname === "/admin.html") {

        if(typeof(Worker) !== "undefined") {

            if(typeof(w) == "undefined") {

              w = new Worker("countdown.js");

            }

            w.onmessage = function(event) {

              if (event.data.done) {

                window.alert("A munka menet lejárt! \nVissza irányítás a fő menüre.");

                window.open("index.html","\_parent");

                // Handle the end of the countdown (e.g., display a message)

                } else {

                    const { minutes, seconds } = event.data;

                    //console.log(`Time remaining: ${minutes}m ${seconds}s`);

                    document.getElementById("work\_time").innerHTML = "0:"+minutes.toString()+":"+seconds.toString();

                    // Update your UI with the remaining time

                }

            };

          } else {

            document.getElementById("work\_time").innerHTML = "Sorry, your browser does not support Web Workers...";

          }

    }

});

## Server-Sent events:

A rólunk oldalon került implementálásra. A grafikon fölött található ami mutatja a napi látogatottságot.

A képen képernyőkép, szöveg, Téglalap, tervezés látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

Ez az adatot egy a szerveren futó php (traffic.php) kód generálja a következő módon.

<?php

header('Content-Type: text/event-stream');

header('Cache-Control: no-cache');

// Random szám

$randomNumber = rand(250, 600);

echo "data: {$randomNumber}\n\n";

flush();

?>

## Geolocation API:

Ez a rész az Étlap (rendeles.html) oldalon található meg, megnyitáskor engedélyt kér a böngésző, a helyzet lekérdezéséhez. Engedélyezés után megjelenik az étterem és a helyzetünk, majd pedig megjelenik egy becsült érték a kiszállításra vonatkozóan.

A képen szöveg, térkép, képernyőkép, étel látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

function showPosition(position) {

    const latitude = position.coords.latitude;

    const longitude = position.coords.longitude;

    document.getElementById("location").innerHTML =

        "Szélesség: " + latitude + "<br>Hosszúság: " + longitude;

    const destinationLat = 46.8962;

    const destinationLon = 19.6690;

    const map = L.map('map').setView([latitude, longitude], 10);

    L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {

        attribution: '&copy; OpenStreetMap contributors'

    }).addTo(map);

    L.marker([latitude, longitude]).addTo(map)

        .bindPopup("Az Ön helyzete").openPopup();

    L.marker([destinationLat, destinationLon]).addTo(map)

        .bindPopup("Legjobb kajálda").openPopup();

    L.polyline([

        [latitude, longitude],

        [destinationLat, destinationLon]

    ], {color: 'blue'}).addTo(map);

}

Többféle error is kezelve lett.

function showError(error) {

    switch(error.code) {

        case error.PERMISSION\_DENIED:

            document.getElementById("location").innerHTML = "Felhasználó elutasította a helymeghatározást.";

            break;

        case error.POSITION\_UNAVAILABLE:

            document.getElementById("location").innerHTML = "A helyzetadatok nem érhetők el.";

            break;

        case error.TIMEOUT:

            document.getElementById("location").innerHTML = "Időtúllépés történt a helyzet meghatározásakor.";

            break;

        case error.UNKNOWN\_ERROR:

            document.getElementById("location").innerHTML = "Ismeretlen hiba történt.";

            break;

    }

}

## Drag and drop API:

A **asztal foglalás** oldalon került implementálásra, ahol a bal felső sarokban található ikon vendég megragadható és a kiválasztott helyre illeszthető. Ennek elnevezése opcionális. Amennyiben változtatni szeretnénk a már elhelyezett vendégen, lehetőségünk van: más helyre illesztésére és a annak **törlésére**.

A képen képernyőkép, Grafika, Grafikus tervezés, embléma látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen. A képen képernyőkép, szöveg látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

A drag és drop eseményeknél csakis a vendég neve kerül átvitelre:  
function drag(ev){

    ev.dataTransfer.setData("text", ev.target.id);

}

## Canvas SVG:

A canvas a ChartJs-hez került felhasználásra. És a weblap több helyen is tartalmaz svg formátumú képeket.

**4.** ChartJS

A ChartJS-t a Rólunk (rolunk.html) oldalon találjuk meg, amin az elmúlt hónapokban a látogatottságot tudjuk nyomon követni.

A esetleges hibákat is kiírjuk a felhasználó számára, többféle eset is kezelve lett.

**A képen szöveg, személy, diagram, képernyőkép látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.**

const chartElement = document.getElementById('myChart');

        if (chartElement) {

            console.log("Canvas elem létezik, inicializáljuk a Chart.js-t");

            const ctx = chartElement.getContext('2d');

            new Chart(ctx, {

                type: 'line',

                data: {

                    labels: ['Szeptember','Október','November','December','Január', 'Február', 'Március', 'Május'],

                    datasets: [{

                        label: 'Látogatottság',

                        data: [280,252,302,310,423,481,468,521],

                        borderColor: 'green',

                        backgroundColor: 'rgba(0, 128, 0, 0.2)',

                        fill: true

                    }]

                },

                options: {

                    responsive: true,

                    maintainAspectRatio: false

                }

            });

        } else {

            console.error("HIBA: Az #myChart elem nem található!");

        }

                        fill: true}] }

1. AJAX

Az ajax az [admin](#_Admin_bejelentkezés) oldalon keresztül a Rendelések kezelése oldalon került implementálásra. Az oldal betöltésénél a szerverről lekéri az adatokat (ajax.js):

window.addEventListener("DOMContentLoaded", function () {

    if (window.location.pathname === "/ajax.html")

        {

            read();

        }

});

Rendelés felvételénél a következő kód fut le, amennyibe valahol üres mező a Rendelés hozzáadás gombra kattint hibát jelez azzal hogy piros keretet kap az adott blokk. Ez a fajta hiba kezelés igaz a többi blokkra az az Rendelések változtatása és Rendelés törlése esetén.

**A képen szöveg, képernyőkép, szoftver, Multimédiás szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.**

1. OOJS

Az oldalt [adminként](#_Admin_bejelentkezés) való bejelentkezés után tudjuk megtalálni a felsorolásban „OOJS alkalmazás” néven. Az oldal megnyitását követően virtuális kisállatokkal tudunk foglalkozni, akikkel tudunk játszani és megtudjuk őket etetni.

A képen szöveg, képernyőkép, Betűtípus, tervezés látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

Az éhségük és a boldogságuk 5-ös értékről indul.

this.hunger = 5;

this.happiness = 5;

Két-két gomb áll a rendelkezésünkre mindkét állatnál. Az „Játék” gombbal lehet velük „játszani”, ami felviszi a boldogságukat, viszont megfognak tőle éhezni (az „Éhség” számláló és a „Boldogság” számláló növekedni fog).

play() {

        this.happiness = Math.min(10, this.happiness + 1);

        this.hunger = Math.min(10, this.hunger + 1);

        this.updateStatus();

    }

Amennyiben az éhség eléri a 8-at vagy annál több lesz, pirossá válik az állat neve és a számlálói.

A képen szöveg, képernyőkép, Betűtípus, tervezés látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

Az „Etetés” gombbal meglehet őket „etetni”, ami leviszi a éhségüket (az „Éhség” számláló csökkenni fog).

feed() {

        this.hunger = Math.max(0, this.hunger - 1);

        this.updateStatus();

}

Az egyik állatnak még egy szuper képessége is van, amit a „játék” gombra kattintva érhetünk el (Console-ban látjuk a kiírását ha megtörténik).

class SpecialPet extends Pet {

    constructor(name, type, specialAbility) {

        super(name, type);

        this.specialAbility = specialAbility;

    }

    play() {

        super.play();

        console.log(`${this.name} ${this.specialAbility}-et csinált! 🎉`);

    }

}

1. React

A képen szöveg, képernyőkép, Mobiltelefon, multimédia látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

Az oldalt [adminként](#_Admin_bejelentkezés) való bejelentkezés után tudjuk megtalálni a felsorolásban „react weboldal” néven. Amikor megnyitjuk az oldalt egy szám generátor fogad minket. A „Generálj számot” feliratú gombra kattintva a nevéből adódóan egy véletlen számot tudunk legenerálni, ami 0 és 1000 között lesz (1-999). Minden alkalommal láthatjuk az éppen legenerált számot a gomb felett.

A szám generátor panel felett két gomb található, ami egy menüként szolgál.

function NumberGenerator() {

  const [number, setNumber] = useState<number>(0);

  const generateNumber = () => {

    setNumber(Math.floor(Math.random() \* 1000));

  };

  return (

    <div className="container">

      <h2 className="title">Szám Generátor</h2>

      <p className="text">Generált szám: {number}</p>

      <button className="button" onClick={generateNumber}>

        Generálj számot

      </button>

    </div>

  );

}

Az egyik opció ami éppen ki van választva az a szám generátor a másik pedig egy színválasztó. A „Színválasztó” gombra kattintva átváltunk a másik panelre.

A képen képernyőkép, szöveg, Operációs rendszer, tervezés látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

A szürke négyzetbe kattintva egy „color picker”-t kapunk ahol az egérrel választhatunk ki egy színt. Miután kiválasztottunk és kikattintunk belőle megjelenik a kiválasztott szín a panel háttérszíneként és megjelenítődik a HEX kódja is.

A képen szöveg, képernyőkép, tervezés látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

function ColorPicker() {

  const [color, setColor] = useState<string>("#ffffff");

  return (

    <div className="container" style={{ backgroundColor: color }}>

      <h3 className="click">Kattins a szürke négyzetbe!</h3>

      <input

        type="color"

        value={color}

        onChange={(e) => setColor(e.target.value)}

        className="color-picker"

      />

      <p className="text">Kiválasztott szín: {color}</p>

    </div>

  );

}

A menük között tetszésszerűen tudunk váltogatni.

export default function App() {

  const [menu, setMenu] = useState<"number" | "color">("number");

  return (

    <div className="app">

      <nav className="nav">

        <button className="button" onClick={() => setMenu("number")}>

          Szám Generátor

        </button>

        <button className="button" onClick={() => setMenu("color")}>

          Színválasztó

        </button>

      </nav>

      <div className="content">

        {menu === "number" ? <NumberGenerator /> : <ColorPicker />}

      </div>

    </div>

  );

}

# Admin bejelentkezés

A képen szöveg, képernyőkép, Betűtípus, Grafikus tervezés látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.Az admin oldal az elérhető a jobb felső sarokban található bejelentkezés fül által. (Felhasználó: admin | jelszó: admin)

A képen képernyőkép, tervezés látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.