

JEGYZŐKÖNYV

Web technológia alapjai

E-Recept weboldal

Készítette: **Kovács István Tamás**

Neptunkód: **AUYLIN**

Dátum: 2024.05.04

Miskolc, 2024.

Tartalomjegyzék

| | |
|--|----|
| Tartalomjegyzék | 2 |
| Bevezetés | 3 |
| 1. Áttekintés | 4 |
| 1.1. Az Online Receptgyűjtemény oldalainak főbb jellemzői: | 4 |
| 1.2. Felhasznált technológiák: | 4 |
| 1.3. Az oldal struktúrája | 5 |
| 2. A weboldal oldalainak kódjának való jellemzése | 6 |
| 2.1. Alapvető weboldal elemek | 6 |
| 2.2. Index.html fontosabb elemei | 7 |
| 2.3. Receptek.html fontosabb elemei | 8 |
| 2.4. Recept.html fontosabb elemei | 12 |
| 2.5. Rate.html fontosabb elemei | 15 |
| 2.5.1 Rate.js | 16 |
| 2.6. Videok.html | 17 |
| 3. NodeJS konfigurálása | 19 |

Bevezetés

A weboldal fő célja az, hogy:

Az online receptgyűjtemény egy weboldal, amelynek fő célja, hogy inspirációt és segítséget nyújtson az embereknek a főzés terén. Az oldal különböző funkciókat kínál, amelyek lehetővé teszik a felhasználók számára, hogy könnyen megtalálják, értékeljék kedvenc receptjeiket.

Recepteket kínáljon: A felhasználók különböző recepteket böngészhetnek az oldalon, így könnyedén megtalálhatják a kedvenc ételeiket vagy új ötleteket szerezhettek a főzéshez. A felhasználók széles választékból válogathatnak recepteket az oldalon. Lehetőségük van kategóriák szerint böngészni, például Tésztafélék, pizzák, desszertek stb.

Értékelés és hozzászólás: A felhasználók értékelhetik a recepteket, megoszthatják véleményüket, tapasztalataikat, és megoszthatják az elkészítésük során szerzett tanulságokat.

Videó bemutatók: A weboldal bemutató videókat is kínál az egyes receptek elkészítéséről, ami segíthet a felhasználóknak abban, hogy könnyebben megértsék az elkészítés folyamatát és technikáját. Ezek a videók lépésről lépésre vezetik végig a felhasználókat az elkészítés folyamatán, segítve őket abban, hogy jobban megértsék a recepteket és könnyebben követhessék azokat.

1. Áttekintés

1.1. Az Online Receptgyűjtemény oldalainak főbb jellemzői:

Főoldal (index.html): A főoldal tartalmazza a legfrissebb recepteket és népszerű ételeket, valamint lehetőséget nyújt a navigációra az alkalmazás többi részére.

Receptek (receptek.html): A receptek oldal lehetővé teszi a felhasználók számára, hogy böngésszenek és keresgéljenek a rendelkezésre álló receptek között. Minden recept kártya formájában jelenik meg, tartalmazva a képet, címet és rövid leírást.

Recept részletei (recept.html): A receptek részletei oldal egy adott recept részletes leírását, hozzávalóit és elkészítési útmutatóját tartalmazza.

Étel értékelése (rate.html): Ez az oldal lehetővé teszi a felhasználók számára, hogy értékeljék az egyes ételeket és hozzászólásokat fűzzenek hozzájuk.

Videók (videok.html): A videók oldal videókat tartalmaz, amelyek bemutatják az egyes ételek elkészítését.

1.2. Felhasznált technológiák:

HTML5 és CSS3: Az oldal szerkezetének és megjelenésének kialakítása.

Bootstrap 4: A reszponzív elrendezés és az előre definiált stílusok használata.

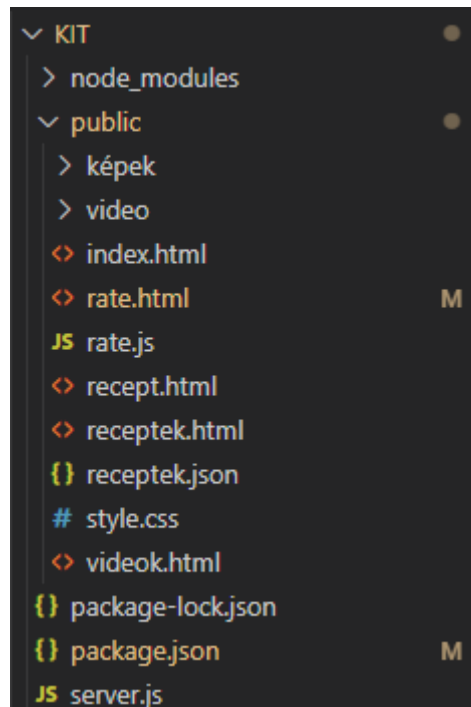
JavaScript (jQuery): Az interaktív funkciók, mint például a képek átváltása és az űrlap validálása.

Ajax: Az aszinkron adatátvitel lehetővé tétele a szerverrel, például dinamikusan betöltött tartalmak, adatok lekérése és frissítése a háttérben.

Node.js: A helyi szerver futtatásához és a háttér logika megvalósításához.

1.3. Az oldal struktúrája

- Gyökérkönyvtár
 - **serve-js:** Ebben a mappában találhatóak a JavaScript fájlok, amelyek valószínűleg a szerver oldali logikát tartalmazzák.
 - **package.json:** A Node.js projekt konfigurációs fájlja, amelyben a projekt függőségei és egyéb beállítások vannak definiálva.
 - **package-lock.json:** A package.json fájl függőségeinek pontos verzióit tartalmazó fájl, hogy azok reprodukálhatóak legyenek.
 - **node_modules:** Ebben a mappában találhatóak a Node.js projekt függőségei, amelyeket a package.json alapján telepítettél.
 - **public:** Ebben a mappában találhatóak azok az állományok, amelyek közvetlenül a böngészőből elérhetőek.
 - Html, js, json, css
 - **képek:** Ebben a mappában találhatóak a képfájlok, amelyek az oldalakon megjelenített képeket tartalmazzák.
 - **videók:** Ebben a mappában találhatóak a videófájlok, amelyek az étel elkészítéséről szólnak.



2. A weboldal oldalainak kódjának való jellemzése

2.1. Alapvető weboldal elemek

Ezek az elemek az oldalak alapstruktúráját alkotják, és minden weboldalon megtalálhatóak

- **Header**

header: A fejléc tartalmazza az oldal címét, valamint a navigációs menüt, amelyen keresztül a felhasználók elérhetik a főoldalt és a receptek oldalt.

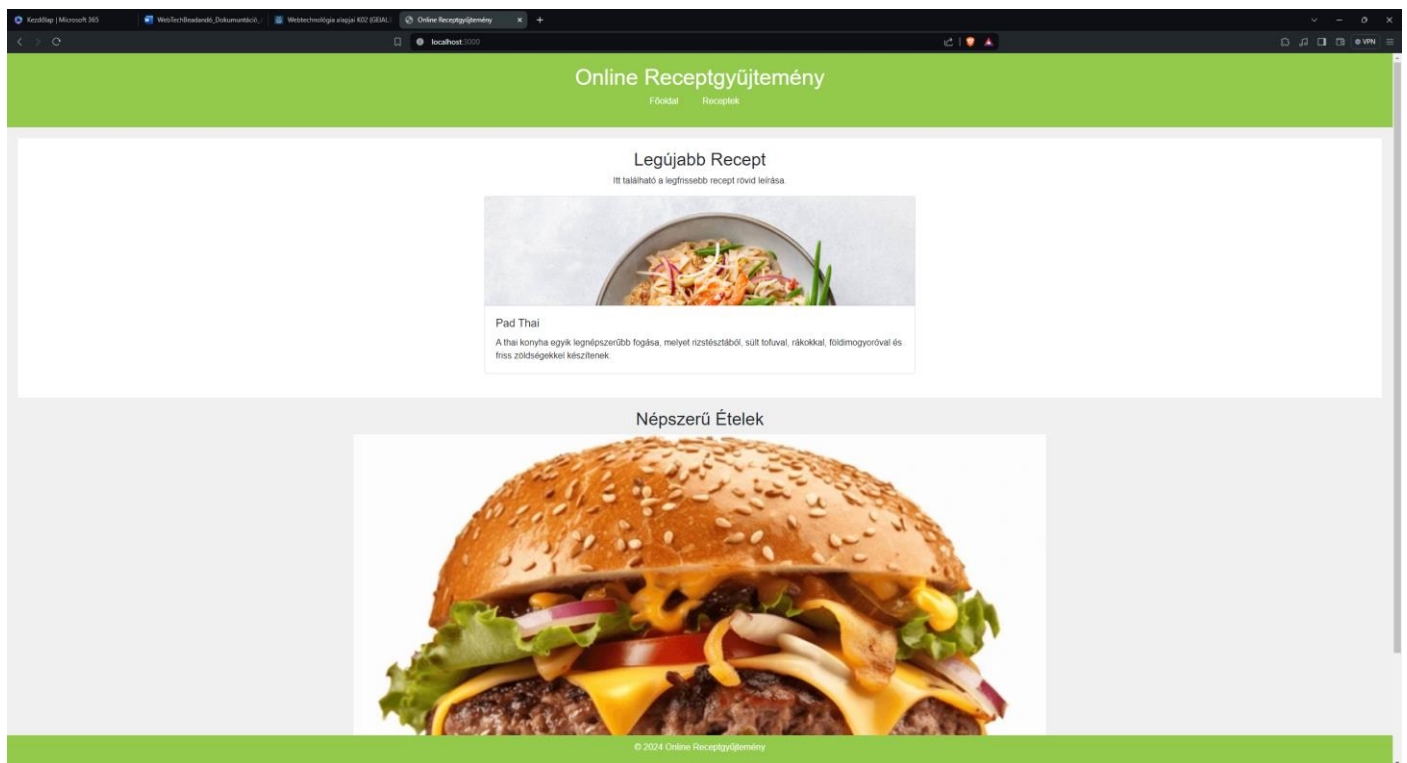
```
<header>
  <h1>Online Receptgyűjtemény</h1>
  <nav>
    <ul>
      <li><a href="index.html">Főoldal</a></li>
      <li><a href="receptek.html">Receptek</a></li>
    </ul>
  </nav>
</header>
```

- **Footer**

footer: A lábléc tartalmazza az oldal alján található információkat, például a szerzői jogi információt vagy egyéb releváns linkeket.

```
<footer>
  <p>&copy; 2024 Online Receptgyűjtemény</p>
</footer>
```

2.2. Index.html fontosabb elemei



- Section:

Ez a szakasz tartalmazza a legújabb receptet, amelyet egy cím, egy rövid leírás és egy kártya formájában mutat be. A recept képe a kártya fejlécében található, míg a recept címe és leírása a kártya testében jelenik meg, de ezekről a kártyákról majd később bővebben kitérek a receptek.html-be

```
<section>
  <h2 style="text-align: center;">Legújabb Recept</h2>
  <p style="text-align: center;">Itt található a legfrissebb recept rövid leírása.</p>
  <div class="col-md-4 mb-4 mx-auto" id="Thai-konyha">
    <div class="card">
      <div class="card-header">
        -----
      </div>
    </div>
  </div>
```

- **Aside:**

Ez az oldalsávban található szakasz, amely a fő tartalomtól elkülönülve jelenik meg. Ideális hely a kiegészítő információk megjelenítésére.

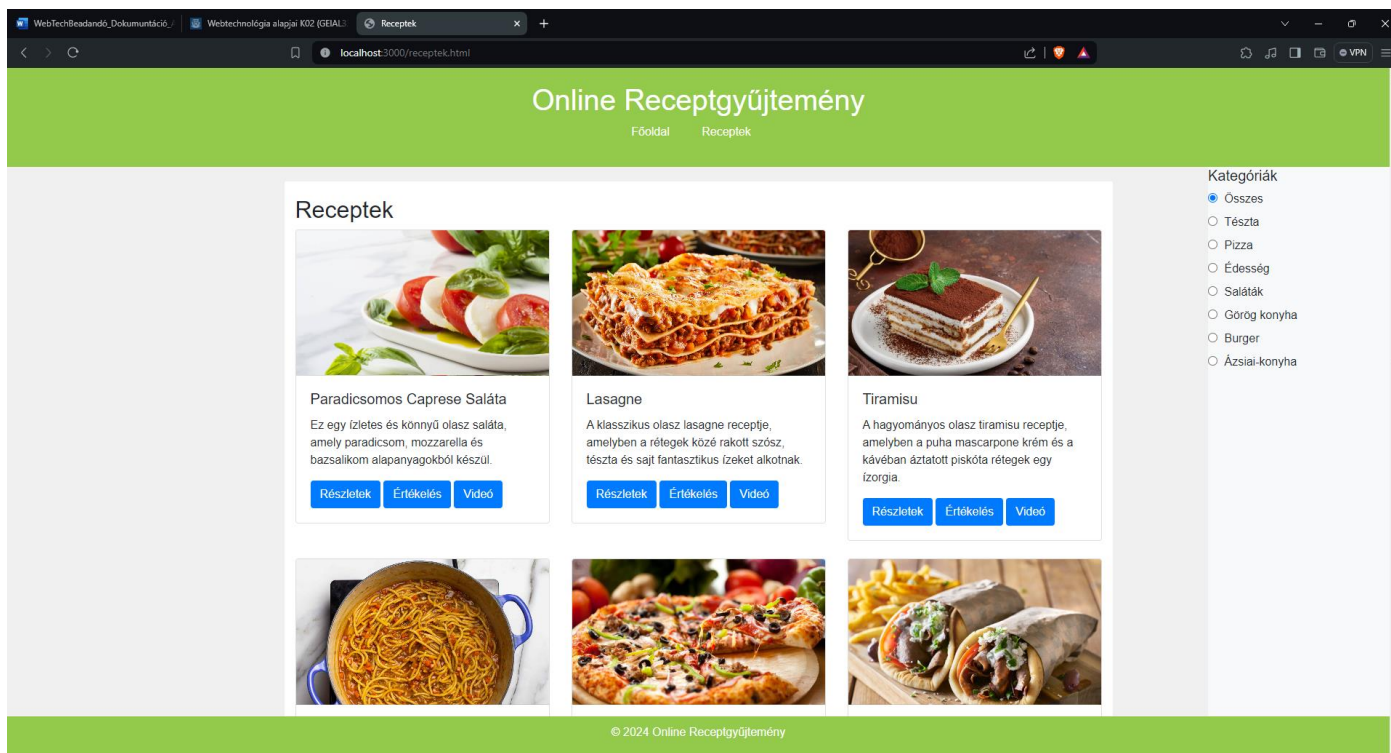
- **Carousel**

Ez is a aside-on belül található. Egy "carousel" egy olyan felhasználói felületi elem, amely lehetővé teszi több kép, szöveg vagy tartalom ciklikus megjelenítését egy weboldalon.

- **carousel-inner:** Ez a div elem tartalmazza a karuzellában megjelenő képeket. Minden képet egy-egy "carousel-item" div elem tartalmaz, amelyek közül az első aktív.
- **carousel-control-prev** és **carousel-control-next:** Ezek a linkek lehetővé teszik a felhasználók számára, hogy előre vagy hátra lépjenek a karuzellában lévő képek között. A "carousel-control-prev-icon" és "carousel-control-next-icon" elemek a nyíl ikonokat jelentik, amelyekkel a felhasználók navigálhatnak a képek között. A "sr-only" osztály azért van ott, hogy a képernyőolvasók számára megfelelő navigációs segítséget nyújtson.

```
<aside>
  <h2 style="text-align: center;">Népszerű Ételek</h2>
  <div id="popular-carousel" class="carousel slide d-block mx-auto" data-ride="carousel">
    <div class="carousel-inner">
      <div class="carousel-item active">
        
      </div>
      <div class="carousel-item">
        
      </div>
      -----
    </div>
  </aside>
```

2.3. Receptek.html fontosabb elemei



Ez az oldal egy online receptgyűjteményt mutat be, amelyben különböző recepteket lehet böngészni és megtekinteni.

- **Side Navigation (Oldalsáv):**

Az oldal jobb oldalán található fixen megjelenő oldalsáv, amely tartalmazza a kategóriák listáját. A kategóriák közötti váltáshoz rádiógombokat használ, amelyekkel a felhasználók szűrhetik a recepteket.

```
<div class="sidenav">
  <h5>Kategóriák</h5>
  <div class="radio-item">
    <input type="radio" name="category" value="osszes" onchange="filterCards(this.value)" checked>
    <label for="osszes">Összes</label>
  </div>
  -----
</div>
```

- **Card (Kártya):**

Minden recept egy kártyában jelenik meg az oldalon.

A kártya három fő részből áll: fejléc, tartalom és gombok.

1. Fejléc:

A kártya fejléce általában a recept képét tartalmazza, amely háttereként szolgál.

A kép lehetőséget ad a felhasználónak a recept vizuális megtekintésére.

2. Tartalom:

A kártya tartalma tartalmazza a recept nevét és rövid leírását.

A recept neve és leírása segít a felhasználónak azonnal azonosítani és megérteni a recept lényegét.

3. Gombok:

A kártya alján található gombok lehetővé teszik a felhasználók számára, hogy további tevékenységeket végezzenek a recepttel kapcsolatban:

"Részletek": A részletes információkra mutató link.

"Értékelés": Az értékelés oldalra mutató link.

"Videó": A recepttel kapcsolatos videóra mutató link.

A kártyák összességében egyszerűen áttekinthetőek és interaktívak, lehetővé téve a felhasználók számára, hogy könnyen böngésszék és kiválasszák a számukra érdekes recepteket az oldalon.

```
<div class="col-md-4 mb-4" id="Saláták">
  <div class="card">
    <div class="card-header" style="background-image: url('képek/caprese.jpg'); background-size: cover; height: 200px;">
    </div>
    <div class="card-body">
      <h5 class="card-title">Paradicsomos Caprese Saláta</h5>
      <p class="card-text">Ez egy ízletes és könnyű olasz saláta, amely paradicsom, mozzarella és bazsalikom alapanyagokból készül.</p>
      <a href="recept.html?etel=caprese" class="btn btn-primary">Részletek</a>
      <a href="rate.html?etel=caprese" class="btn btn-primary">Értékelés</a>
      <a href="videok.html?etel=caprese" class="btn btn-primary">Videó</a>
    </div>
  </div>
</div>
```

JavaScript kód:

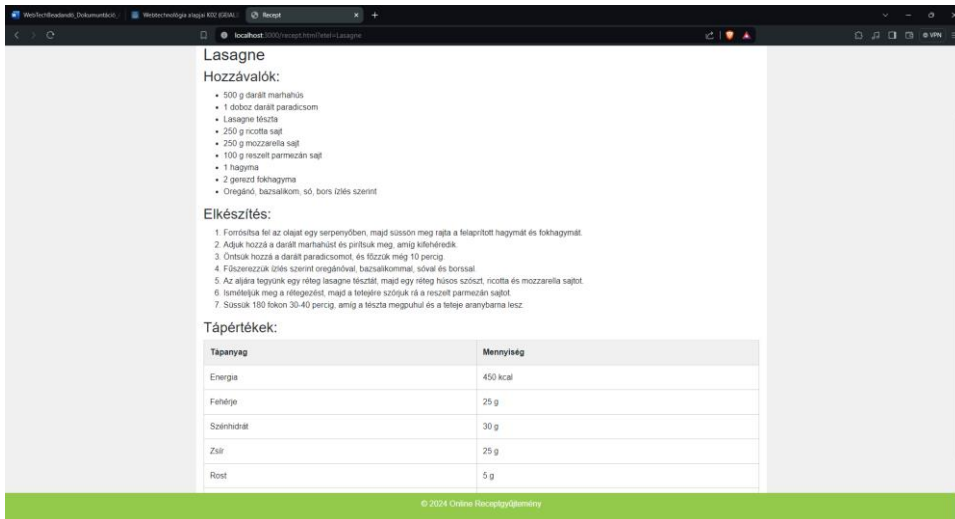
A kártyákra vonatkozó egérmozgás eseményeket kezeli, amelyek megváltoztatják a kártyák pozícióját a hover eseményekre.

A kategóriák szűrését végző JavaScript függvény, amely reagál az oldalsávban lévő rádiógombok változására, és megjeleníti vagy elrejtí a megfelelő kategóriákhoz tartozó recepteket.

```
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script>
    $(document).ready(function(){
        $(".card").mouseenter(function(){
            $(this).css("transform", "translateY(-10px)");
        });
        $(".card").mouseleave(function(){
            $(this).css("transform", "translateY(0)");
        });
    });

    function filterCards(category) {
        var cards = document.getElementsByClassName('col-md-4 mb-4');
        for (var i = 0; i < cards.length; i++) {
            if(category.toLowerCase() == "osszes"){
                cards[i].style.display = 'block';
            }
            else{
                if (cards[i].id.toLowerCase().includes(category.toLowerCase())) {
                    cards[i].style.display = 'block';
                } else {
                    cards[i].style.display = 'none';
                }
            }
        }
    }
</script>
```

2.4. Recept.html fontosabb elemei



- **Section (Szakasz):**

A recept tartalmát tartalmazza.

- recipe-title id-vel ellátott cím megjeleníti a recept nevét.
- hozzavalok id-vel ellátott rész tartalmazza a hozzávalók listáját.
- elkeszites id-vel ellátott rész tartalmazza az elkészítés lépéseinek listáját.
- tapertekek id-vel ellátott rész tartalmazza a tápértékek táblázatát.

```
<section class="container">
  <h2 id="recipe-title"></h2>

  <div id="hozzavalok">
    <h3>Hozzávalók:</h3>
    <ul></ul>
  </div>

  <div id="elkeszites">
    <h3>Elkészítés:</h3>
    <ol></ol>
  </div>

  <div id="tapertekek">
    <h3>Tápértékek:</h3>
    <table class="table">
      <thead>
        <tr>
          <th>Tápanyag</th>
          <th>Mennyiség</th>
        </tr>
      </thead>
      <tbody></tbody>
    </table>
  </div>
</section>
```

Így néz ki a json fájl egy része:

```
[
  {
    "nev": "caprese",
    "kategoria": "Saláták",
    "hozzavalok": [
      "Friss paradicsom",
      "Mozzarella sajt",
      "Friss bazsalikom levelek",
      "Extra szűz olívaolaj",
      "Balsamicó krém",
      "Só és frissen őrölt fekete bors ízlés szerint"
    ],
    "elkeszites": [
      "Mosson meg alaposan és vágja fel a paradicsomokat és a mozzarella vékony szeletekre.",
      "Rendezze a paradicsom- és mozzarellaszeleteket egy tányérra váltakozva.",
      "Szórjon rá friss bazsalikomleveleket.",
      "Csepegtessen az egészre extra szűz olívaolajat és balsamicó krémet.",
      "Sózza és borsozza ízlés szerint.",
      "Frissen tálalva azonnal fogyasztandó."
    ],
    "tapertek": {
      "Energia": "250 kcal",
      "Fehérje": "12 g",
      "Szénhidrát": "8 g",
      "Zsír": "18 g",
      "Rost": "3 g",
      "Nátrium": "500 mg",
      "Koleszterin": "25 mg",
      "Cukor": "4 g"
    }
  },
]
```

Script (Javascript kód):

```
<script>
    var urlParams = new URLSearchParams(window.location.search);
    var etel = urlParams.get('etel');

    $(document).ready(function() {
        $.getJSON("receptek.json", function(data) {
            var recept = data.find(function(item) {
                return item.nev === etel;
            });
            if (recept) {
                $("#recipe-title").text(recept.nev);
                var hozzavalokList = $("#hozzavalok ul");
                var elkeszitesList = $("#elkeszites ol");
                var tapertekekTable = $("#tapertekek tbody");
                $.each(recept.hozzavalok, function(index, item) {
                    hozzavalokList.append("<li>" + item + "</li>");
                });
                $.each(recept.elkeszites, function(index, item) {
                    elkeszitesList.append("<li>" + item + "</li>");
                });
                $.each(recept.tapertekek, function(key, value) {
                    tapertekekTable.append("<tr><td>" + key + "</td><td>" + value + "</td></tr>");
                });
            } else {
                $("#recipe-title").text("Nincs ilyen recept");
            }
        });
    });
</script>
```

- Az oldal tartalmát dinamikusan tölti be az adatok alapján.
- Az URL-ben kapott étel nevével lekéri az adatokat az receptek.json fájlból.
- Megjeleníti a recept nevét, hozzávalóit, elkészítési lépéseit és tápértékeit az oldalon.

Ez a metódus aszinkron módon küld kéréseket a szervernek és JSON formátumban várja a választ. Az \$.getJSON() metódus maga hajtja végre az AJAX kérést, a háttérben, a megadott URL címre. A válasz JSON formátumú adat, amit a callback függvény (a második argumentum) dolgoz fel.

2.5. Rate.html fontosabb elemei

Étel értékelése

Főoldal Receptek

Értékelés

Értékelj az alábbi ételt Lasagne

1. Mennyire finom?

3 - Közepes

2. Elkészíthetőség:

3 - Közepes

3. Elkészíthetőség:

3 - Közepes

4. Megjegyzés (max. 1000 karakter):

Kérjük ügyeljen:

- Ne használjon speciális karaktereket(@&#)

5. Egyéb kérdések: elkészítené megint?

☐

6. Mikor készítettél el az ételt?

éé.éé. hh. nn.

7. Milyen a kedvenc ételod színe?

Ajánlani

☐ Igen ☐ Nem

Értékelés elküldése

© 2024 Online Receptgyűjtemény

Értékelő űrlap: Az oldalon található űrlap lehetővé teszi a felhasználók számára, hogy értékeljék az ételt. Az űrlap tartalmazza a következő elemeket:

- A kiértékelendő étel neve (#étel-nev), amely dinamikusan jelenik meg az oldalon.
- A finom, élvezhető és drágaság értékelésére szolgáló adatlisták.
- Egy megjegyzés mező (textarea), amelyben a felhasználó megoszthatja véleményét.
- Egy "elkészítené megint" checkbox (input type="checkbox").
- Egy dátumválasztó (input type="date").
- Egy színpicker (input type="color").
- Ajánlásra szolgáló rádiógombok (input type="radio").
- Végül egy "Értékelés elküldése" gomb (button type="submit").

Űrlap validálás: A megjegyzés mező maximum 1000 karaktert fogad el, és a felhasználót figyelmezteti, ha túllépi ezt a határt. Emellett a felhasználókat kérjük, hogy ne használjanak speciális karaktereket a megjegyzésben.

JavaScript funkciók: A JavaScript kód segítségével dinamikusan jelenítik meg az étel nevét az oldalon. Emellett van egy külön JavaScript fájl (rate.js), amely valószínűleg további funkciókat és eseménykezelést tartalmaz az értékelő űrlaphoz.

2.5.1 Rate.js

A rate.js egy JavaScript fájl, amely az értékelő űrlap validálását és az értékelés elküldését végzi. Néhány fontos jellemzője:

- **jQuery használata:** A kód jQuery-t használ az eseménykezeléshez és a DOM manipulációhoz.
- **Űrlap validálás:** Az űrlap beküldését figyeli meg a \$('#rating-form').submit() eseménykezelő. Megvizsgálja a megjegyzés mező tartalmát és ellenőrzi, hogy a karakterek száma ne legyen több mint 1000, valamint hogy ne tartalmazzon speciális karaktereket.
- **Hibaüzenetek:** Ha a validálás során hibát talál, a megfelelő hibaüzenetet jeleníti meg a felhasználónak, és az űrlap beküldését megakadályozza (event.preventDefault()). Az invalid mezőket is kiemeli a Bootstrap stílusok segítségével.
- **Sikeres értékelés esetén:** Ha az értékelés sikeresen beküldésre került, egy üzenetet jelenít meg.
- **validateTextarea() függvény:** Ez a függvény külön validálja a megjegyzés mezőt. Ellenőrzi, hogy a megjegyzés hossza nem haladja-e meg a megengedett maximumot, és hogy nem tartalmaz-e speciális karaktereket. Ha hiba van, megjeleníti a megfelelő hibaüzenetet és kiemeli a mezőt. Ez a függvény az oninput eseményre van beállítva a megjegyzés mezőn.

```
$(document).ready(function() {
    $('#rating-form').submit(function(event) {
        var megjegyzes = $('#megjegyzes').val();
        var maxChars = 1000;
        var regex = /[!@#$$%^&*()_+~\-=\[\]\{\};':"\|,.<>\?]+/;
        var error = '';

        if (megjegyzes.length > maxChars) {
            error = 'A megjegyzés maximum 1000 karakter lehet';
        } else if (regex.test(megjegyzes)) {
            error = 'A megjegyzés nem tartalmazhat speciális karaktereket';
        }

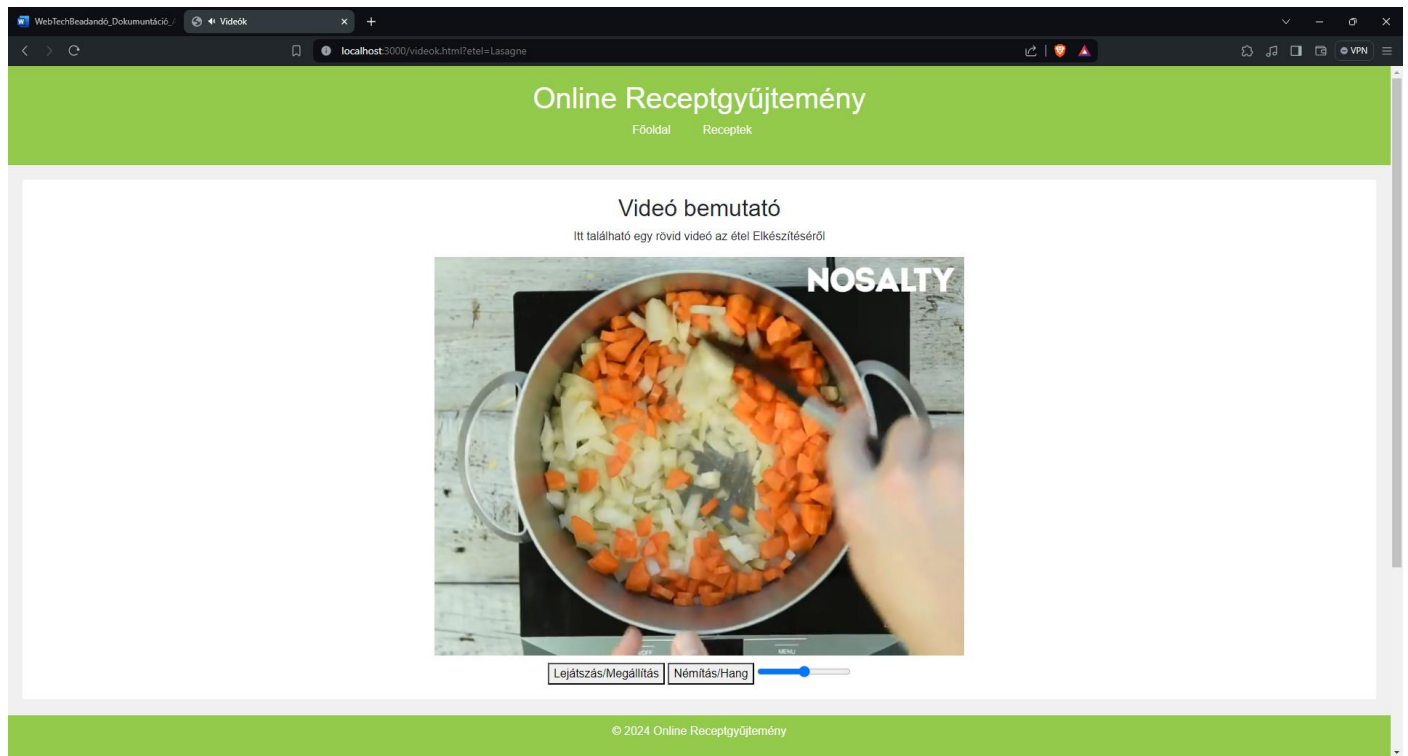
        if (error) {
            $('#megjegyzes-error').text(error);
            $('#megjegyzes').addClass('is-invalid');
            event.preventDefault();
        } else {
            $('#megjegyzes-error').text('');
            $('#megjegyzes').removeClass('is-invalid');
            alert('Értékelés sikeresen elküldve!');
            window.location.href = 'receptek.html';
        }
    });
});

function validateTextarea(textarea) {
    var maxChars = 1000;
    var text = textarea.value;
    var errorDiv = document.getElementById('megjegyzes-error');

    if (text.length > maxChars) {
        errorDiv.textContent = 'A megjegyzés maximum 1000 karakter lehet';
        textarea.classList.add('is-invalid');
    } else {
        errorDiv.textContent = '';
        textarea.classList.remove('is-invalid');
    }

    var regex = /[!@#$$%^&*()_+~\-=\[\]\{\};':"\|,.<>\?]+/;
    if (regex.test(text)) {
        errorDiv.textContent = 'A megjegyzés nem tartalmazhat speciális karaktereket';
        textarea.classList.add('is-invalid');
    }
}
```


2.6. Videok.html



AZ ÖSSZES VIDEO A NOSALTY CSATORNÁRÁJOL VAN

<https://www.youtube.com/@nosalty>

A `videok.html` egy olyan HTML oldal, amely egy videó bemutatót tartalmaz az étel elkészítéséről. Néhány fontos jellemzője:

Videó lejátszása: Az oldal középpontjában egy HTML5 `<video>` elem található, amelynek beágyazott kontrollerei vannak. A videó URL-je dinamikusan generálódik az URL paraméter alapján (étel). Ha nincs megadva étel, akkor egy alapértelmezett videót jelenít meg.

Videó vezérlők: A videó lejátszását és hangvezérlését lehetővé tevő gombok (Lejátszás/Megállítás és Némítés/Hang), valamint a hangerőszabályzó (`input type="range"`) a videó alatt találhatóak. Ezek a funkciók JavaScript segítségével vannak kezelve.

```
<section>
  <h2 style="text-align: center;">Videó bemutató</h2>
  <p style="text-align: center;">Itt található egy rövid videó az étel Elkészítéséről</p>
  <video id="videoPlayer" controls style="height: 40%; width: 40%; display: block; margin: 0 auto;">
    A böngésző nem támogatja a videót.
  </video>
  <div style="text-align: center; margin-top: 10px;">
    <button onclick="playPause()">Lejátszás/Megállítás</button>
    <button onclick="muteUnmute()">Némítés/Hang</button>
    <input type="range" id="volumeControl" min="0" max="1" step="0.1" onchange="changeVolume()">
  </div>
</section>
```

JavaScript funkciók: A JavaScript részben számos funkció található:

- **playPause():** Az aktuális lejátszás/megállítás funkciót kezeli.
- **muteUnmute():** A hang némítását és újraaktiválását végzi.
- **changeVolume():** A hangerőszabályzó állítása alapján módosítja a videó hangerőjét.
- A **window.onload** esemény figyeli az oldal betöltését, és beállítja a megfelelő videóforrást az étel paraméter alapján.

```
<script>
window.onload = function() {
    var etel = getUrlParameter('etel');
    var videoPlayer = document.getElementById('videoPlayer');

    if (etel) {
        var videoSource = "video/" + etel.replace(/ /g, "_") + ".mp4";

        videoPlayer.src = videoSource;

        videoPlayer.play();
    }

    function getUrlParameter(name) {
        name = name.replace(/\[/, '\\[').replace(/\\]/, '\\]');
        var regex = new RegExp('[\\?&]' + name + '=(^&#)*');
        var results = regex.exec(location.search);
        return results === null ? '' : decodeURIComponent(results[1].replace(/\+/g, ' '));
    };
}

var videoPlayer = document.getElementById('videoPlayer');
var volumeControl = document.getElementById('volumeControl');

function playPause() {
    if (videoPlayer.paused) {
        videoPlayer.play();
    } else {
        videoPlayer.pause();
    }
}

function muteUnmute() {
    if (videoPlayer.muted) {
        videoPlayer.muted = false;
    } else {
        videoPlayer.muted = true;
    }
}

function changeVolume() {
    videoPlayer.volume = volumeControl.value;
}
</script>
```

Az oldal így egy dinamikus videót bemutatót nyújt az étel elkészítéséről, ahol a felhasználók a videó lejátszását vezérelhetik, valamint beállíthatják a hangerejét.

3. NodeJS konfigurálása

1. server.js létrehozása

Hozz létre egy server.js fájlt a projekt gyökérkönyvtárában, és írd bele a következő kódot:

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

const app = express();
const port = 3000;

app.use(express.static(path.join(__dirname, 'public')));

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

app.use((req, res, next) => {
  res.status(404).sendFile(path.join(__dirname, 'public', '404.html'));
});

app.use((err, req, res, next) => {
  console.error(err.stack);
  res.status(400).sendFile(path.join(__dirname, 'public', '400.html'));
});

app.listen(port, () => {
  console.log(`A szerver fut a http://localhost:\${port} címen.`);
});
```

2. package.json létrehozása és konfigurálása

Nyiss egy terminált a projekt gyökérkönyvtárában, majd futtasd az alábbi parancsot a package.json létrehozásához:

```
npm init -y
```

Ezután módosítsd a package.json fájlt, hogy a következőképpen nézzen ki:

3. Szükséges csomagok telepítése

A csomagok telepítéséhez futtasd az alábbi parancsot a terminálban:

```
npm install
```

4. Szerver indítása

A szerver indításához futtasd az alábbi parancsot:

```
npm start
```

Ezután a szerver elindul a localhost:3000-on, és a böngészőben láthatod a "Hello World!" üzenetet.

Ha bármilyen további funkciót vagy beállítást szeretnél hozzáadni a szerverhez, módosíthatod a server.js fájlt a kívánt módon.