

# FlavorWave Webalkalmazás Dokumentáció

Darida Gergő, Böröndi Ádám, Szitási Zalán

2025. május 13.

## Tartalomjegyzék

1	Bevezetés .....	3
2	Felhasználói Felületek .....	3
3	Adatbázisterv .....	3
3.1	Táblák Leírása .....	3
3.2	Adatbázis Diagram .....	5
4	Funkciók Részletes Leírása .....	5
4.1	Felhasználók .....	5
4.2	Ételek .....	5
4.3	Extrák .....	5
4.4	Extrák Opciói .....	5
4.5	Rendelések .....	6
5	Fejlesztési Mérföldkövek .....	6
6	Előrehaladás Nyomon Követése .....	6
7	Feladatok Részletes Lebontása .....	6
7.1	Zalán: Frontend Fejlesztés .....	6
7.2	Gergő: Admin és Dolgozói Felületek .....	7
7.2.1	Adatbázis – Étel Tábla .....	7
7.2.2	Ételkezelő Felület Fejlesztése .....	7
7.2.3	Jogosultságkezelés és Szerkesztési Funkciók .....	8
7.2.4	Rendelések Kezelése .....	8
7.2.5	Valós Idejű Állapotfrissítések .....	9
7.2.6	Visszajelzési Rendszer Fejlesztése és Megjelenítése .....	9
7.2.7	Kód Optimalizálása és Karbantartása .....	9
7.3	Ádám: Backend Fejlesztés .....	10
7.3.1	Adatbázis Tervezés és Struktúra Létrehozása .....	10
7.3.2	Felhasználói Regisztráció és Bejelentkezés .....	10
7.3.3	Profilom .....	11
7.3.4	Kosár és Rendelések Kezelése .....	12
7.3.5	Rendelés Megtekintése .....	12
7.3.6	Ételek és Kategóriák Kezelése .....	12

7.3.7	Vélemények Kezelése és Megjelenítése .....	12
7.3.8	Backend Biztonság és Optimalizálás .....	13
8	Technikai Megvalósítás .....	13
8.1	Frontend .....	13
8.2	Backend .....	14
8.3	Oldal Bemutatása .....	14
8.3.1	Navigációs Sáv .....	14
8.3.2	Kosárban Lévo Termékek Számának Lekérdezése .....	15
8.3.3	Főoldal .....	17
8.4	Regisztráció .....	19
8.5	Bejelentkezés .....	20
8.6	Profilom .....	21
8.7	Rendelés .....	22
8.8	Rendelés Megtekintése .....	23
8.9	Visszajelzések .....	24
9	Kosárkezelő Rendszer Dokumentáció .....	25
9.1	Bevezetés .....	25
9.2	Rendszeráttekintés .....	25
9.3	Adatbázis-séma .....	25
9.4	Munkamenet- és Sütikezelés .....	25
9.5	Szkriptek Részletes Bemutatása .....	26
9.5.1	kosarba_rakas.php .....	27
9.5.2	kosarbol_torles.php .....	27
9.5.3	kosar_uritese.php .....	27
9.5.4	kosar_mod.php .....	28
9.6	Biztonsági Szempontok .....	28
9.7	Összegzés .....	28
10	Használt Technológiák .....	29
11	Következtetés .....	30

## 1 Bevezetés

A FlavorWave egy képzeletbeli gyorséttermi lánc weboldalát valósítja meg, amely átfogó online ételrendelési élményt kínál. A felhasználók regisztrálhatnak fiókot, bejelentkezhetnek,

nyomon követhetik rendeléseiket, és részt vehetnek egy hűségprogramban, ahol pontokat gyűjthetnek kedvezményekre vagy ingyenes ételekre. Az étlap részletes, tartalmazza az ételek nevét, árát, összetevőit, kalóriaértékeit, valamint akciókat. A rendelési folyamat egyszerű, lehetővé téve az ételek testreszabását (pl. öntetek kiválasztása, összetevők eltávolítása). A rendelés végén a felhasználók e-mailben kapnak visszaigazolást a részletekkel, végösszeggel, kedvezményekkel és várható kiszállítási idővel. További funkciók közé tartozik a korábbi rendelések áttekintése, újrendelés, valamint bankkártyás és online fizetési lehetőségek.

## 2 Felhasználói Felületek

A webalkalmazás a következő felületeket tartalmazza:

- Regisztrációs Oldal: Felhasználók személyes adataikat megadva hozhatnak létre fiókot.
- Bejelentkezési Oldal: Meglévő fiókkal léphetnek be a rendelési funkciókhoz.
- Főoldal: Aktuális akciókat, ajánlatokat és navigációs elemeket jelenít meg.
- Étlap: Kategóriákra bontva (ételek, italok, desszertek), részletes leírásokkal és árakkal.
- Rendelési Felület: Ételválasztás, testreszabás, összesítés és elkészítési idő megtekintése.
- Értékelési Felület: Rendelés után visszajelzés adható az ételről és kiszállításról.
- Admin és Dolgozói Felületek: Ételek, rendelések és jogosultságok kezelése.

## 3 Adatbázis-terv

Az adatbázis kilenc táblából áll, amelyek a rendszer működését támogatják. Az alábbiakban a táblák leírása található:

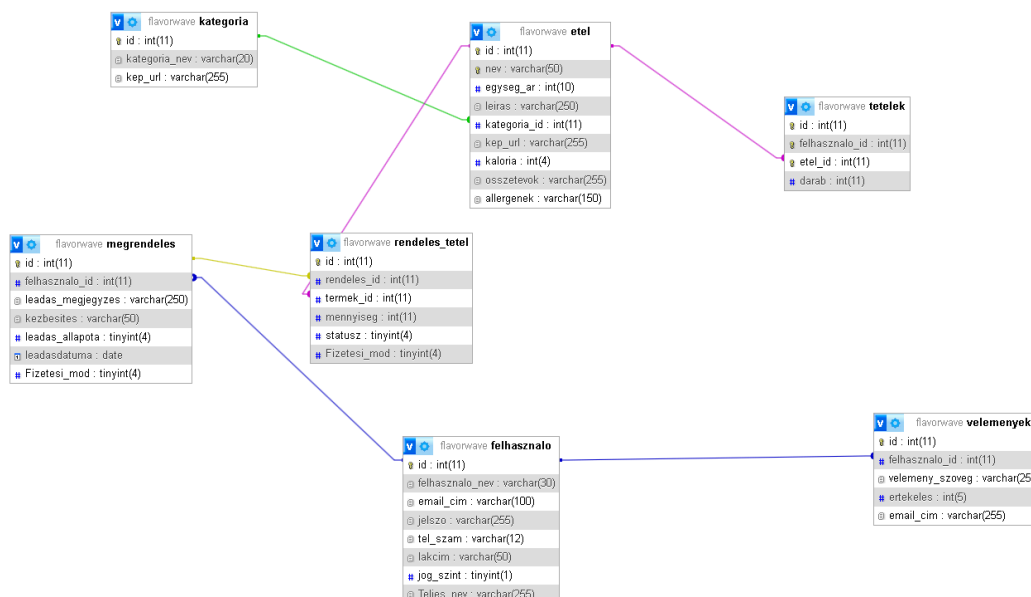
### 3.1 Táblák Leírása

- etel (Étel):
  - ID (Elsődleges Kulcs): Egyedi ételazonosító.
  - nev: Étél neve (pl. „Margherita Pizza”).
  - egyseg\_ar: Egységár (pl. 1500 Ft).
  - leiras: Rövid leírás (pl. összetevők).
  - kategoria\_id (Idegen Kulcs): Hivatkozik a kategoria táblára.
  - kep\_url: Az étel képének URL-je.
  - kaloria: Az étel kalóriaszáma.
  - osszetevok: Az étel összetevői.
  - allergenek: Az étel allergénjei.
- felhasználó (Felhasználó):
  - ID (Elsődleges kulcs): Egyedi felhasználó azonosító.
  - felhasználó\_nev: Egyedi felhasználónév.

- email\_cim: E-mail cím.
- jelszo: Titkosított jelszó.
- tel\_szam: Telefonszám.
- lakcim: Szállítási cím.
- jog\_szint: Jogosultsági szint.
- Teljes\_nev: A felhasználó teljes neve.
- kategoria (Kategória):
  - ID (Elsődleges Kulcs): Egyedi kategóriaazonosító.
  - kategoria\_nev: Kategória neve (pl. „Pizza”).
- megrendeles (Rendelés):
  - ID (Elsődleges Kulcs): Egyedi rendelésazonosító.
  - felhasznalo\_id: A felhasználó azonosítója..
  - leadasmegjegyzes: Opcionális megjegyzés.
  - kezbesites: Szállítási információk.
  - leadasallapota: Rendelés állapota (pl. „Feldolgozás alatt”).
  - leadasdatum: Rendelés leadásának dátuma és ideje.
  - Fizetesi\_mod: A rendelés fizetési módja.
- tetelek (Rendelési Tételek):
  - ID (Elsődleges Kulcs): Egyedi tételazonosító.
  - felhasznalo\_id (Idegen Kulcs): A felhasználó azonosítója.
  - etel\_id (Idegen Kulcs): Hivatkozik az etel táblára.
  - darab: Rendelt mennyiség.
- velemenyek (Vélemények):
  - ID (Elsődleges Kulcs): Egyedi véleményazonosító.
  - felhasznalo\_id: A felhasználó azonosítója.
  - velemeny\_szoveg: Szöveges visszajelzés.
  - ertekeles: Értékelés (1-5 csillag)
  - email\_cim: A felhasználó email címe.
- rendeles\_tetel
  - ID (Elsődleges Kulcs): Egyedi rendelésazonosító.
  - rendeles\_id: A rendelés azonosítója.
  - termék\_id: A termék azonosítója.
  - mennyiseg: A rendelés mennyisége.
  - statusz: A rendelés státusza

- Fizetesi\_mod: A rendelés fizetési módja.

## 3.2 Adatbázis Diagram



## 4 Funkciók Részletes Leírása

A rendszer a következő műveleteket támogatja:

### 4.1 Felhasználók

- **Interakció:** Regisztráció során a felhasználó megadja nevét, e-mail címét, jelszavát, telefonszámát és lakcímét. Bejelentkezéskor e-mail és jelszó szükséges.
- **Üzleti Logika:** Ellenőrzi az e-mail egyediségét és a jelszó biztonsági követelményeit. Bejelentkezéskor validálja a hitelesítő adatokat.
- **Adatbázis Műveletek:** Új felhasználó rögzítése a felhasznalo táblában, profilfrissítés, bejelentkezési adatok ellenőrzése.

### 4.2 Ételek

- **Interakció:** A felhasználó böngészi az étlapot, ahol ételek neve, ára és leírása látható.
- **Üzleti Logika:** Dinamikus étlapkijelzés kategóriák szerint, rendelési végösszeg kiszámítása.
- **Adatbázis Műveletek:** Ételek lekérdezése az etel táblából, rendelés során ételazonosítók mentése a tetelek táblába.

### 4.3 Extrák

- Interakció: A felhasználó extra feltéteket (pl. sajt, szósz) adhat az ételekhez.
- Üzleti Logika: A feltétek növelik az árat, csak kompatibilis feltétek választhatók.
- Adatbázis Műveletek: Feltétek lekérdezése az extrafeltet táblából, kiválasztások mentése a feltet\_opcio táblába.

### 4.4 Extrák Opciói

- Interakció: Az extrák és ételek közötti kapcsolatokat kezeli, rögzíti a kiválasztott feltéteket.
- Üzleti Logika: Biztosítja, hogy csak az ételhez tartozó extrák rendelhetők.
- Adatbázis Műveletek: A feltet\_opcio tábla frissítése a kiválasztott feltétekkel.

### 4.5 Rendelések

- Interakció: Ételek kiválasztása, mennyiség megadása, rendelés leadása szállítási adatokkal.
- Üzleti Logika: Ellenőrzi az ételek elérhetőségét, feltétek helyességét, végösszeg kiszámítása, e-mail visszaigazolás küldése.
- Adatbázis Műveletek: Rendelés rögzítése a megrendeles táblában, tételek mentése a tetelek táblába, e-mail státusz frissítése.

## 5 Fejlesztési Mérföldkövek

Feladat	Kezdés	Befejezés	Felelős
Téma meghatározása	2024.09.03	2024.09.10	Csapat
Adatbázis megtervezése	2024.09.10	2024.10.01	Csapat
Első dokumentáció	2024.09.03	2024.10.25	Csapat
Felhasználói felületek tervezése	2024.09.17	2024.10.22	Gergő
Frontend fejlesztés	2024.10.26	2025.03.24	Zalán
Backend fejlesztés	2024.10.26	2025.04.01	Ádám
Dizájn véglegesítése	2024.11.26	2025.04.10	Zalán
Tesztelés	2025.03.01	2025.04.30	Csapat
Végső átadás	2025.05.01	2025.05.10	Csapat

## 6 Előrehaladás Nyomon Követése

- Menedzselt Nyomkövetés: Trello táblák: „Fejlesztésre Kiválasztva”, „Fejlesztés Alatt”, „Átnézésre Vár”, „Tesztelve”, „Kész”.
- Programozói Nyomkövetés: GitHub pull requestekkel és kódellenőrzéssel.
- Dokumentációs Nyomkövetés: Folyamatos frissítés, késések okainak dokumentálása.

## 7 Feladatok Részletes Lebontása

### 7.1 Zalán: Frontend Fejlesztés, Design, Főoldal, Menü oldalak

- Főoldal és Dizájn (Határidő: 2025.02.28): Letisztult, reszponzív főoldal akciókkal.
- Étlap Oldalak (Határidő: 2025.03.31): Kategorizált étlap reszponzív elrendezéssel.
- Navigációs Sáv és Lábléc (Határidő: 2024.11.30): Intuitív navigáció és informatív lábléc.
- Kosár Funkcionalitás (Határidő: 2025.01.25): Dinamikus kosárfrissítés JavaScripttel.
- Rendelés Véglegesítése (Határidő: 2025.02.10): Összegzés és backend integráció.
- Reszponzív Dizájn (Határidő: 2025.03.15): Keresztplatform kompatibilitás.
- Végso Optimalizálás (Határidő: 2025.05.15): Teljesítményjavítás és hibajavítás.

### 7.2 Gergő: Admin és Dolgozói Felületek

- Étel Tábla (Határidő: 2024.11.15): Étel tábla képekkel és tápértékekkel.
- Ételkezelő Felület (Határidő: 2024.12.05): Admin felület ételek kezelésére.
- Jogosultságkezelés (Határidő: 2024.12.20): Szerepkörök és ételszerkesztés.
- Rendelések Kezelése (Határidő: 2025.01.15): Dolgozói felület rendeléskezeléshez.
- Valós Idejű Frissítések (Határidő: 2025.01.25): AJAX rendelésállapot-frissítések.
- Visszajelzési Rendszer (Határidő: 2025.02.20): Vélemények gyűjtése reCAPTCHAval.
- Kód Optimalizálása (Határidő: 2025.03.10): Hatékonyság és biztonság növelése.

#### 7.2.1 Adatbázis – Étel Tábla

Határidő: 2024. november 15.

A cél az étel tábla tervezése és integrálása, hogy hatékonyan tárolja az ételek adatait, beleértve a képek elérési útját és a tápérték-információkat. A tábla az admin felülettel szorosan együttműködik, biztosítva az ételek kezelését.

Oszlop neve	Típus	Leírás
<u>id</u>	INT	Egyedi azonosító, automatikusan generált
<u>nev</u>	<u>VARCHAR(100)</u>	Az étel neve, pl. "Csokis süti"
<u>egyseg_ar</u>	<u>DECIMAL(10,2)</u>	Az étel ára, pl. 599.00
<u>leiras</u>	<u>TEXT</u>	Az étel leírása
<u>kategoria_id</u>	INT	Az étel kategóriájának azonosítója
<u>kep_url</u>	<u>VARCHAR(255)</u>	A feltöltött kép elérési útja
<u>kaloria</u>	INT	Kalóriatartalom
<u>osszetevok</u>	<u>TEXT</u>	Összetevők listája
<u>allergenek</u>	<u>TEXT</u>	Allergének listája

Magyarázat: Az étel tábla minden szükséges adatot tárol, a kategoria\_id pedig egy másik táblára hivatkozik az ételek csoportosításához. A képfeltöltés során egyedi azonosítókat használtam az ékezetes karakterek problémáinak elkerülésére.

### 7.2.2 Ételkezelő Felület Fejlesztésey

Határidő: 2024. december 5.

Az adminisztrátori felület alapvető funkcionálisát PHP és MySQL segítségével fejlesztették ki, hogy az adminisztrátorok könnyen kezelhessék az ételadatbázist. A cél egy egyszerű, átlátható és biztonságos felület létrehozása, amely lehetővé teszi az ételek hozzáadását, szerkesztését, törlését és képek feltöltését. A kód modulárisan épült fel, hogy a jövőben könnyen bővíthető legyen, és ügyeltek a felhasználói visszajelzésekre (pl. hibaüzenetek) a hatékony adminisztráció érdekében.

Funkciók:

- **Ételek hozzáadása:** Az adminisztrátor megadhatja az étel nevét, árát, leírását, kategóriáját, kalóriatartalmát, összetevőit, allergénjeit és egy képet.
- **Képfeltöltés:** JPG, PNG vagy GIF formátumú képek feltöltése, maximum 5 MB méretben.
- **Validáció:** A kötelező mezők ellenőrzése és a fájlformátumok szűrése.

**Példa: Étel hozzáadása**



```

if (isset($_POST['operation']) && $_POST['operation'] == 'add') {
    if (empty($_POST['nev']) || empty($_POST['egyseg_ar']) || empty($_POST['leiras']) || empty($_POST['kategoria_id']) || empty($_POST['kaloria']) || empty($_POST['osszetevok']) || empty($_POST['allergenek'])) {
        echo '<div class="alert alert-danger">Minden mező kitöltése kötelező!</div>';
    } else {
        if (isset($_FILES['kepek_url']) && $_FILES['kepek_url']['error'] == 0) {
            $allowed = ['jpg', 'jpeg', 'png', 'gif'];
            $filename = $_FILES['kepek_url']['name'];
            $filetype = pathinfo($filename, PATHINFO_EXTENSION);
            if (!in_array(strtolower($filetype), $allowed)) {
                echo '<div class="alert alert-danger">Csak JPG, JPEG, PNG, GIF fájlokat tölthet fel!</div>';
            } else {
                $target_dir = "Uploads/" . $_POST['kategoria_id'] . "/";
                $unique_id = uniqid();
                $target_file = $target_dir . $unique_id . '_' . basename($_POST['nev'] . '.' . $filetype);

                if (move_uploaded_file($_FILES["kepek_url"]["tmp_name"], $target_file)) {
                    $sql = "INSERT INTO etel (nev, egyseg_ar, leiras, kategoria_id, kep_url, kaloria, osszetevok, allergenek) VALUES (?, ?, ?, ?, ?, ?, ?, ?)";
                    $stmt = $conn->prepare($sql);
                    $stmt->bind_param("sdssiiss", $_POST['nev'], $_POST['egyseg_ar'], $_POST['leiras'], $_POST['kategoria_id'], $target_file, $_POST['kaloria'], $_POST['osszetevok'], $_POST['allergenek']);

                    if ($stmt->execute()) {
                        echo '<div class="alert alert-success">Étel sikeresen hozzáadva!</div>';
                    } else {
                        echo '<div class="alert alert-danger">Hiba az étel hozzáadása során!</div>';
                    }
                } else {
                    echo '<div class="alert alert-danger">Hiba a fájl feltöltésekor!</div>';
                }
            }
        } else {
            echo '<div class="alert alert-danger">Kérem, töltsön fel egy képet!</div>';
        }
    }
}

```

Aktiválja  
Aktiválja a v

Magyarázat: A kód lehetővé teszi egy új étel, például egy „Csokis süti” hozzáadását. Ellenőrzi a bemeneteket, validálja a képformátumot, és egyedi azonosítóval menti a fájlt, hogy elkerülje az ékezetes karakterek problémáit. A készített utasítások biztosítják az SQL injekciók elleni védelmet.

### 7.2.3 Jogosultságkezelés és Szerkesztési Funkciók

Határidő: 2024. december 20.

A cél az admin felület bővítése jogosultságkezelési és ételszerkesztési funkciókkal, hogy az adminisztrátorok hatékonyan kezelhessék a felhasználói jogosultságokat és az ételek adatait. A jogosultságkezelés lehetővé teszi a dolgozói szerepkörök kiosztását, míg az ételszerkesztés a meglévő ételek adatainak módosítását biztosítja.

Funkciók:

- Jogosultságkezelés: Dolgozói jogosultságok adása vagy visszavonása.
- Étel szerkesztése: Meglévő ételek adatainak módosítása.
- Felhasználói visszajelzések: Siker- és hibaüzenetek a műveletek eredményéről.

### Példa: Étel szerkesztése

```
if (isset($_POST['operation']) && $_POST['operation'] == 'edit') {
    $etel_id = $_POST['etel_id'];
    if (empty($_POST['nev']) || empty($_POST['egyseg_ar']) || empty($_POST['leiras']) || empty($_POST['kategoria_id'])) {
        echo '<div class="alert alert-danger">Minden mező kitöltése kötelező!</div>';
    } else {
        $sql = "UPDATE etel SET nev = ?, egyseg_ar = ?, leiras = ?, kategoria_id = ?, kaloria = ?, osszetevok = ?, allergenek = ? WHERE id = ?";
        $stmt = $conn->prepare($sql);
        $stmt->bind_param("sdssiiss", $_POST['nev'], $_POST['egyseg_ar'], $_POST['leiras'], $_POST['kategoria_id'], $_POST['kaloria'], $_POST['osszetevok'], $_POST['allergenek'], $etel_id);
        if ($stmt->execute()) {
            echo '<div class="alert alert-success">Étel sikeresen szerkesztve!</div>';
        } else {
            echo '<div class="alert alert-danger">Hiba az étel szerkesztése során!</div>';
        }
    }
}
```

Magyarázat: A kód egy meglévő étel adatainak frissítését végzi az `etel_id` alapján. A bemenetek validálása és a készített utasítások használata biztosítja a biztonságos működést.

## 7.2.4 Rendelések Kezelése

Határidő: 2025. január 15.

A dolgozói felület fejlesztésének célja egy olyan rendszer létrehozása, amely támogatja a rendelések hatékony kezelését a gyorséttermi működés során. A felület lehetővé teszi a rendelések listázását, azok állapotának kezelését és a rendelési adatok megtekintését.

Funkciók:

- Rendelések listázása: Aktuális rendelések táblázatos megjelenítése.
- Részletek megtekintése: Ételek, mennyiségek és árak megtekintése.
- Állapotkezelés: Rendelések állapotának manuális frissítése.

### Példa: Rendelések listázása

```

$sql = "SELECT m.id, m.felhasznalo_id, m.leadas_datuma, m.leadas_allapota, u.email
        FROM megrendeles m
        JOIN felhasznalok u ON m.felhasznalo_id = u.id
        WHERE m.leadas_allapota IN ('friss', 'folyamatban', 'kesz')";
$stmt = $conn->prepare($sql);
$stmt->execute();
$result = $stmt->get_result();
while ($row = $result->fetch_assoc()) {
    echo '<tr>';
    echo '<td>' . htmlspecialchars($row['id']) . '</td>';
    echo '<td>' . htmlspecialchars($row['email']) . '</td>';
    echo '<td>' . htmlspecialchars($row['leadas_datuma']) . '</td>';
    echo '<td>' . htmlspecialchars($row['leadas_allapota']) . '</td>';
    echo '<td><button class="btn btn-primary" onclick="updateStatus(' . $row['id'] . ')">Frissítés
</button></td>';
    echo '</tr>';
}

```

Magyarázat: A kód lekéri az aktuális rendeléseket, és táblázatos formában jeleníti meg őket, beleértve a rendelés azonosítóját, a felhasználó e-mail címét és az állapotot. A HTML-kódolás véd az XSS támadások ellen.

### 7.2.5 Valós Idejű Állapotfrissítések

Határidő: 2025. január 25.

A dolgozói felületet AJAX-alapú frissítésekkel bővítették, hogy a rendelések állapota valós időben módosulhasson, így növelve a dolgozók hatékonyságát.

Funkciók:

- Valós idejű állapotfrissítés: Rendelések állapotának módosítása AJAX kéréssel.
- Visszajelzések: Siker- és hibaüzenetek a műveletek eredményéről.

#### Példa: Rendelés állapotának frissítése

```

if (isset($_POST['action']) && $_POST['action'] == 'update_order_status') {
    $order_id = $_POST['order_id'];
    $new_status = $_POST['new_status'];
    $sql = "UPDATE megrendeles SET leadas_allapota = ? WHERE id = ?";
    $stmt = $conn->prepare($sql);
    $stmt->bind_param("si", $new_status, $order_id);
    if ($stmt->execute()) {
        echo json_encode(['success' => true, 'message' => 'Rendelés állapota frissítve!']);
    } else {
        echo json_encode(['success' => false, 'message' => 'Hiba a rendelés frissítése során!']);
    }
}

```

Magyarázat: A kód AJAX kéréssel frissíti a rendelés állapotát, és JSON válaszban ad visszajelzést. Ez lehetővé teszi a valós idejű frissítéseket anélkül, hogy az oldal újratöltődne.

### 7.2.6 Visszajelzési Rendszer Fejlesztése és Megjelenítése

Határidő: 2025. február 20.

## Dokumentáció 12

A visszajelzési rendszer célja a felhasználói vélemények gyűjtése és megjelenítése, Google reCAPTCHA-val védve a spam ellen. A vélemények átláthatóan jelennek meg az oldalon.

Funkciók:

- Értékelés beküldése: Csillagos értékelés és szöveges visszajelzés megadása.
- Vélemények megjelenítése: E-mail címmel, értékeléssel és dátummal.
- Spam védelem: Google reCAPTCHA integráció.

### Példa: Visszajelzés beküldése, lekérdezése

```
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $recaptcha_response = $_POST['g-recaptcha-response'];
    $recaptcha_url = 'https://www.google.com/recaptcha/api/siteverify';
    $recaptcha_secret = 'YOUR_SECRET_KEY';
    $recaptcha = file_get_contents($recaptcha_url . '?secret=' . $recaptcha_secret . '&response=' .
    $recaptcha_response);
    $recaptcha = json_decode($recaptcha);
    if ($recaptcha->success == false) {
        echo '<div class="alert alert-danger">reCAPTCHA ellenőrzés sikertelen. Próbálja újra!</div>';
    } else {
        $ertekeles = $_POST['megelegedettseg'];
        $visszajelzes = htmlspecialchars($_POST['visszajelzes']);
        $felhasznalo_id = $_SESSION['user_id'];
        $sql = "INSERT INTO visszajelzes (felhasznalo_id, ertekeles, szoveg, datum) VALUES (?, ?, ?,
        NOW())";
        $stmt = $conn->prepare($sql);
        $stmt->bind_param("iiss", $felhasznalo_id, $ertekeles, $visszajelzes);
        if ($stmt->execute()) {
            echo '<div class="alert alert-success">Köszönjük a visszajelzést!</div>';
        } else {
            echo '<div class="alert alert-danger">Hiba a visszajelzés beküldése során!</div>';
        }
    }
}
```

Magyarázat: A kód ellenőrzi a reCAPTCHA-t, majd menti a felhasználó értékelését és szöveges visszajelzését. A lekérdező kód HTML-kódolással véd az XSS támadások ellen.

```
$sql = "SELECT v.ertekeles, v.szoveg, v.datum, u.email
        FROM visszajelzes v
        JOIN felhasznalok u ON v.felhasznalo_id = u.id
        ORDER BY v.datum DESC";
$stmt = $conn->prepare($sql);
$stmt->execute();
$result = $stmt->get_result();
while ($row = $result->fetch_assoc()) {
    echo '<div class="review">';
    echo '<p><strong>' . htmlspecialchars($row['email']) . '</strong> (' . htmlspecialchars($row['d
    atum']) . ')</p>';
    echo '<p>Értékelés: ' . str_repeat('★', $row['ertekeles']) . str_repeat('☆', 5 - $row['ertekele
    s']) . '</p>';
    echo '<p>' . htmlspecialchars($row['szoveg']) . '</p>';
    echo '</div>';
}
```

Magyarázat: A kód lekéri és megjeleníti a visszajelzéseket, HTML-kódolással védve az XSS támadások ellen. A csillagos értékelés vizuálisan vonzó és könnyen értelmezhető.

### 7.2.7 Kód Optimalizálása és Karbantartása

Határidő: 2025. március 10.

A cél a kód hatékonyságának és karbantarthatóságának biztosítása. A PHP és JavaScript kód átvizsgálása, redundanciák eltávolítása, és a rendszer bővíthetőségének biztosítása.

Funkciók:

- Kódoptimalizálás: Redundáns kódok eltávolítása és lekérdezések optimalizálása.
- Biztonsági intézkedések: SQL injekciók, XSS támadások és spam elleni védelem.
- Dokumentáció: A kód jól dokumentált, hogy a jövőbeli fejlesztések egyszerűbbek legyenek.

Példa: Biztonsági Intézkedések:

- SQL injekció elleni védelem: Készített utasítások minden adatbázis-művelethez.
- XSS védelem: Felhasználói bevitel kódolása `htmlspecialchars()` függvénnyel.
- Fájlfeltöltés validáció: Csak JPG, PNG és GIF képek, max. 5 MB.
- reCAPTCHA: A visszajelzési űrlap Google reCAPTCHA-val védett.

Magyarázat: Ezek az intézkedések biztosítják, hogy a rendszer biztonságos és megbízható legyen éles környezetben.

## 7.3 Ádám: Backend Fejlesztés

- Adatbázis Struktúra (Határidő: 2024.11.15): Kilenc táblás séma tervezése.
- Regisztráció és Bejelentkezés (Határidő: 2024.12.01): Biztonságos hitelesítés.
- Kosár és Rendelés Backend (Határidő: 2025.01.10): Dinamikus kosár és rendelésfeldolgozás.
- Étlap és Kategóriák (Határidő: 2024.12.15): Étel- és kategóriakezelés.
- Vélemények (Határidő: 2025.02.01): Visszajelzések tárolása és megjelenítése.
- Biztonság és Optimalizálás (Határidő: 2025.03.10): Adatbázis biztonság és lekérdezés optimalizálás.

### 7.3.1 Adatbázis Tervezés és Struktúra Létrehozása

Határidő: 2024. november 15.

A weboldal backend működésének alapja az adatbázis, amelyet XAMPP környezetben, MySQL használatával hoztak létre. A csapat közösen felelős az adatbázis logikai felépítéséért, amely kilenc táblából áll. Az egyes táblák különböző funkciókat látnak el: az ételek és feltétek kezelését, a felhasználói regisztrációt, valamint a rendelések és kosár működését. A táblák közötti relációkat úgy alakították ki, hogy a tárolt adatok egységesen és hatékonyan legyenek kezelhetők.

### 7.3.2 Felhasználói Regisztráció és Bejelentkezés

Határidő: 2024. december 1.

A felhasználó tábla adatkezelési funkcióinak megvalósítása, amely a felhasználói adatok tárolását biztosítja. A regisztráció során a felhasználók adatait titkosítva mentik, a jelszavakat `password_hash()` függvénnyel titkosítják. A bejelentkezés során a titkosított jelszavakat `password_verify()` segítségével ellenőrzik.

## Regisztráció

Felhasználónév:

Email cím:

Jelszó (min. 8 karakter, szám és betű):

Telefonszám (06 vagy +36 kezdettel):

**Regisztráció**

Már van fiókod? [Bejelentkezés](#)

**Bejelentkezés**

Felhasználónév:

Jelszó:

☐ Nem vagyok robot  reCAPTCHA  
Adatvédelem - Általános Szerződési Feltételek

**Bejelentkezés**

Nincs fiókod? [Regisztrálj most!](#)

#### Funkciók:

- Adatbázis Kapcsolat: MySQL adatbázishoz csatlakozás.
- Űrlapfeldolgozás és Validáció: Felhasználónév, e-mail, jelszó, telefonszám ellenőrzése.
- Felhasználó Regisztráció: Titkosított jelszóval mentés a felhasználó táblába.
- Email Küldés: PHPMailer HTML e-mailek küldésére Gmail SMTP-n keresztül.
- reCAPTCHA Ellenőrzés: Google reCAPTCHA a robotok kiszűrésére.
- Kosár Egyesítése: Vendégkosár és felhasználói kosár egyesítése bejelentkezés után.

#### Biztonsági Szempontok:

- Jelszó titkosítás `password_hash()` és `password_verify()` használatával.

- SQL injekció elleni védelem előkészített lekérdezésekkel.
- Google reCAPTCHA integráció a brute-force támadások ellen.
- Tranzakciókezelés a kosár egyesítése során.

### 7.3.3 Profilom

Határidő: 2025. február 1.

A profilom.php fájl a felhasználói profil kezelését biztosítja, lehetővé téve a személyes adatok megtekintését és módosítását.

Kép helye: Profil Alert Üzenetek (image40.png)

Figure 15: Profil Alert Üzenetek

Kép helye: Profil Szerkesztés (image41.png)

Figure 16: Profil Szerkesztés

Kép helye: Profil Interaktív Szerkesztés (image43.png)

Figure 17: Profil Interaktív Szerkesztés Főbb

Funkciók:

- Felhasználói adatok betöltése a felhasználó táblából.
- Profil szerkesztés interaktív beviteli mezőkkel.
- Kliensoldali validáció e-mailre és telefonszámra.
- Megerősítő modal ablak a mentés előtt.

Biztonsági Szempontok:

- SQL injekció elleni védelem előkészített lekérdezésekkel.
- XSS védelem htmlspecialchars() függvénnyel.
- Session kezelés a hozzáférés korlátozására.

### 7.3.4 Kosár és Rendelések Kezelése

Határidő: 2025. január 10.

A rendelési folyamat kulcsfontosságú elemeinek implementálása. A megrendelés tábla az összesített rendelési adatokat tárolja, a tetelek tábla a rendelések egyes tételeinek részleteit tartalmazza. A kosár dinamikus szerkesztése és eltávolítása optimalizált.



## Rendelés véglegesítése

Teljes név:

Szállítási cím (Irányítószám, Utca, Házsám):

Telefonszám:

Fizetési mód

☐ Készpénz

☐ Bankkártya

Megjegyzés (opcionális)

Rendelés véglegesítése

### Rendelés összegzése

Sajtburger (23 db)	34500 Ft
Dupla sajtburger (3 db)	6000 Ft
Dupla húsos burger (6 db)	13200 Ft
Csirke burger (2 db)	3600 Ft
Standard burger (2 db)	2800 Ft
Baconos burger (2 db)	4000 Ft
<b>Összesen:</b>	<b>64100 Ft</b>

Funkciók:

- Kosáradatok lekérdezése és megjelenítése.
- Szállítási adatok validálása és rendelés mentése.
- Email visszaigazolás PHPMailer-rel.
- Tranzakciókezelés az adatkonzisztencia érdekében.

### 7.3.5 Rendelés Megtekintése

Határidő: 2025. február 1.

A `rendeles_megtekint.php` fájl a korábbi rendelések áttekintését biztosítja, rendelési azonosító, dátum, állapot, fizetési mód és tételek megjelenítésével.

Főbb Funkciók:

- Rendelések lekérdezése a megrendelés és tetelek táblákból.
- Adatok csoportosítása rendelési azonosító szerint.
- Reszponzív kártyák a rendelések megjelenítéséhez.

### 7.3.6 Ételek és Kategóriák Kezelése

Határidő: 2024. december 15.

Az ételekkel kapcsolatos adatok kezelése több táblát érint: etel, categoria, tetelek, extrafeltet. Az etel tábla az alapvető ételeket, a categoria tábla a besorolásokat tartalmazza.

### 7.3.7 Vélemények Kezelése és Megjelenítése

Határidő: 2025. február 1.

A vélemények tábla a felhasználói visszajelzések tárolását biztosítja, amelyek az adminisztrációs felületen és az étlapon is megjelennek.

### 7.3.8 Backend Biztonság és Optimalizálás

Határidő: 2025. március 10.

- Adatbázis Biztonság: Titkosított jelszavak, HTTPS kommunikáció.
- Lekérdezések Optimalizálása: Indexek használata, relációk hatékony kezelése.

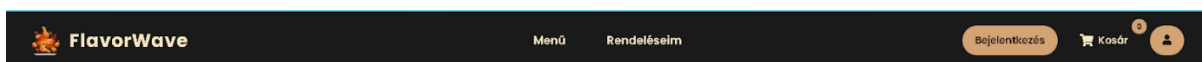
## 8 Technikai Megvalósítás

Az alkalmazás PHP, MySQL, HTML, CSS és JavaScript technológiákra épül, Bootstrap 5-tel reszponzív dizájnhoz.

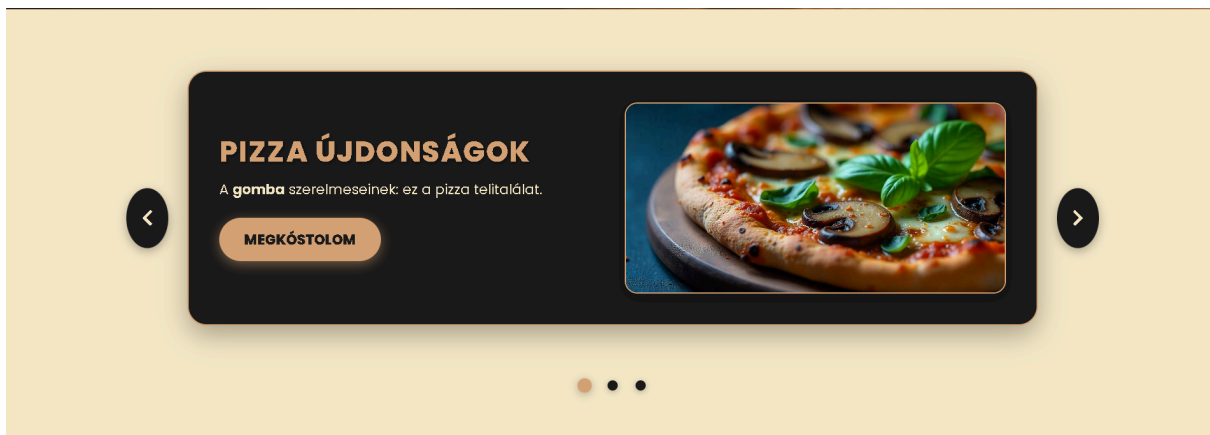
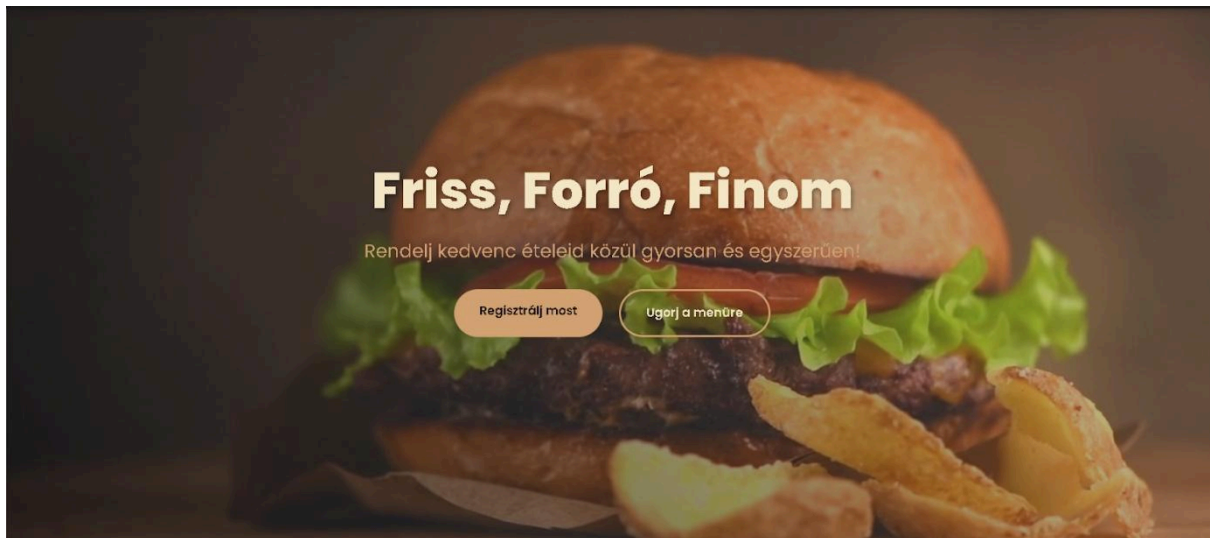
### 8.1 Frontend

- Navigációs Sáv: Logó, menüpontok, kosár tételszámmal, dinamikus bejelentkezés/kijelentkezés. Mobilnézetben hamburger menü.
- Főoldal: Videós hero szekció, újdonságok és heti ajánlatok csúszkái, rendelési lépések és előnyök szekciói.
- Étlap és Rendelési Oldalak: Kategorizált ételek testreszabási opciókkal, dinamikus kosárfrissítés.
- Reszponzív Dizájn: Mobil, tablet és asztali eszközök kompatibilitása.

Navbar:



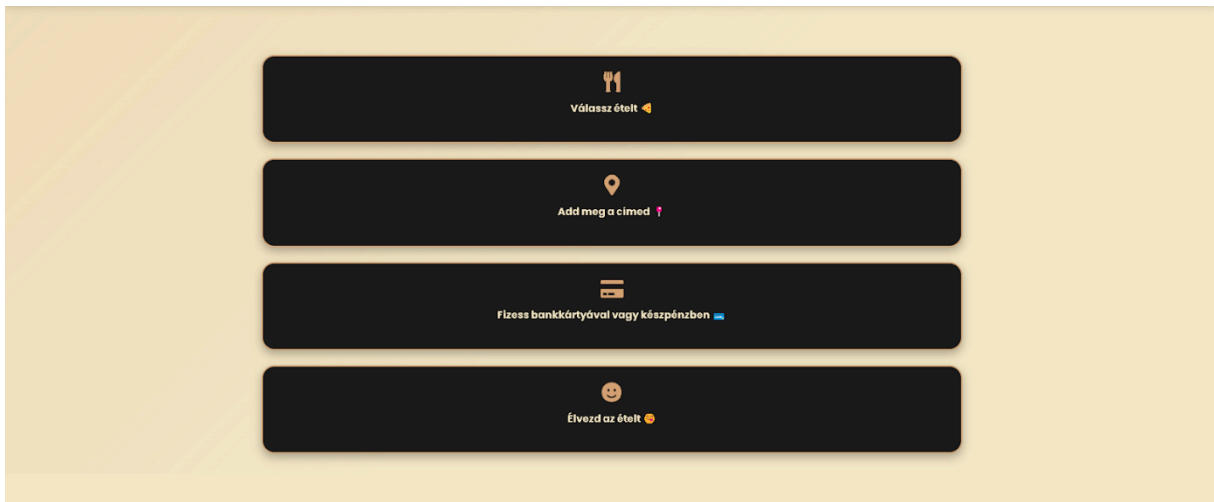
Hero szekció:



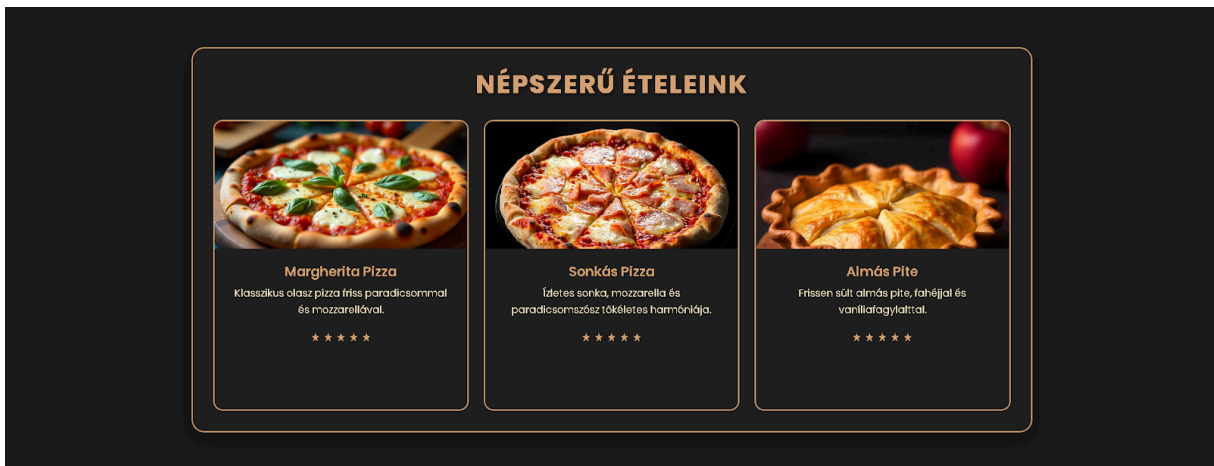
Heti ajánlat vége rész:



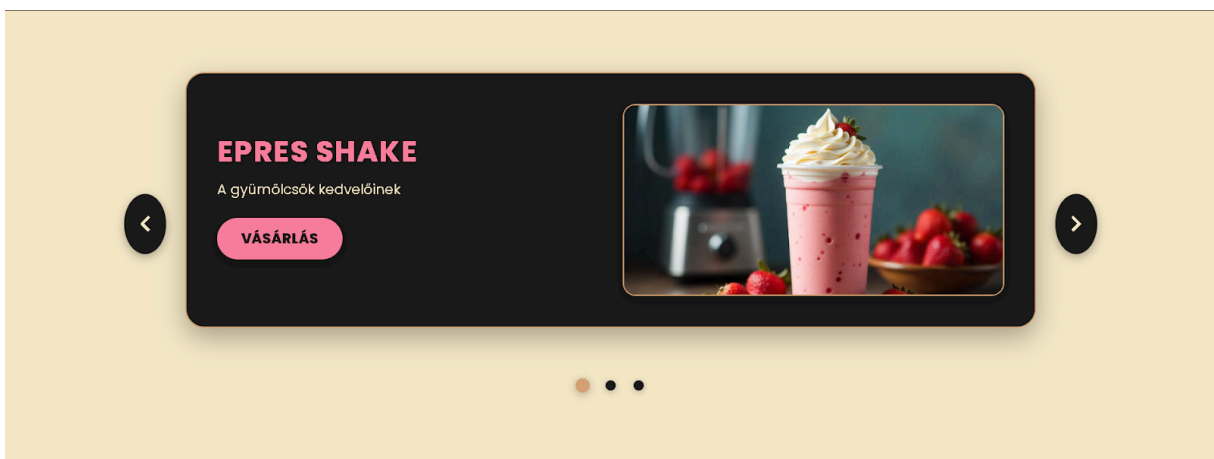
Rendelési lépések rész:



Népszerű ételeink rész:



Shaker rész:



Vélemény küldése be nem jelentkezett felhasználóknak:

Kérjük, **jelentkezz be**, hogy véleményt írhas!

Vélemény küldése bejelentkezett felhasználóknak:

KÜLDD EL A VÉLEMÉNYED!

Miért minket válassz rész:

#### MIÉRT MINKET VÁLASSZ?



Villámgyors kiszállítás



Friss alapanyagok



Egyedi ízek

Footer:

[Rólunk](#) [Kapcsolat](#) [Adatvédelem](#)



© 2024 FlavorWave - Minden jog fenntartva.

## 8.2 Backend

- Adatbázis Kapcsolat: MySQLi előkészített lekérdezésekkel a biztonságos műveletekhez.
- Felhasználó Hitelesítés: password\_hash() és password\_verify() a jelszavak biztonságos kezeléséhez.
- E-mail Értesítések: PHPMailer HTML e-mailek küldésére Gmail SMTP-n keresztül.
- Biztonság: SQL injekció elleni védelem, XSS védelem htmlspecialchars() függvénnyel, HTTPS kommunikáció.

## 8.3 Oldal Bemutatása

### 8.3.1 Navigációs Sáv

A FlavorWave weboldal navigációs sávja modern, letisztult felületet kínál. A bal oldalon a logó, középen a menüpontok (pl. „Menü”, „Rendeléseim”), a jobb oldalon a bejelentkezés/kijelentkezés, kosár ikon tételszámmal és profil ikon található. Mobilnézetben hamburger menü nyílik.

Backend Működés:

- Dinamikus bejelentkezés/kijelentkezés gomb.
- Kosár tételszám dinamikus frissítése.
- Admin/dolgozó menüpontok jogosultság alapján.

Szükséges Előfeltételek:

- Session kezelés (session\_start()).
- Adatbázis kapcsolat (\$conn).

### 8.3.2 Kosárban Lévo Termékek Számának Lekérdezése

A getItemCount(\$conn) függvény számolja a kosárban lévő termékeket:

- Bejelentkezett felhasználó: Adatbázisból olvassa a darabszámot.
- Vendég felhasználó: Session-ben tárolt kosarat nézi.

### 8.3.3 Főoldal

A főoldal modern bemutatkozó felület több szekcióval:

- Hero Szakasz: Videós szekció „Friss, Forró, Finom” felirattal, regisztrációs vagy rendelési gombokkal.
- Újdonságok Szakasz: Csúszka három ajánlattal (pizza, hamburger, hot dog).
- Heti Ajánlat Szakasz: Visszaszámláló és pizzaképek csúszkái.
- Rendelési Lépések: Négy ikon a rendelési folyamat bemutatására.
- Népszerű Ételek Galéria: Képek az ételekről.
- Shake Szakasz: Három shake ajánlat csúszkával.
- Visszajelzés Szakasz: Bejelentkezett vagy bejelentkezés nélküli visszajelzési lehetőség.
- Miért Minket Válassz Szakasz: Platform előnyeinek bemutatása.

## 8.4 Regisztráció

A regisztracio.php fájl a regisztrációs funkcionalitást valósítja meg, kezelve az űrlap adatainak validálását, mentését és e-mail visszaigazolás küldését.

Főbb Funkciók:

- Adatbázis kapcsolat MySQL-hez.
- Űrlapvalidáció (felhasználónév, e-mail, jelszó, telefonszám).
- Jelszó titkosítás password\_hash() segítségével.
- HTML e-mail küldés PHPMailer-rel.
- Reszponzív Bootstrap 5 űrlap jelszó láthatóság váltóval.

Biztonsági Szempontok:

- SQL injekció elleni védelem előkészített lekérdezésekkel.

- XSS védelem htmlspecialchars() függvénnyel.
- Titkosított SMTP kapcsolat (STARTTLS).

## 8.5 Bejelentkezés

A bejelentkezes.php fájl a bejelentkezési funkcionalitást kezeli, Google reCAPTCHA-val és kosár egyesítéssel.

Főbb Funkciók:

- reCAPTCHA ellenőrzés a robotok kiszűrésére.
- Jelszó validálás password\_verify() segítségével.
- Vendégkosár és felhasználói kosár egyesítése tranzakciókezeléssel.
- Reszponzív Bootstrap 5 űrlap.

## 8.6 Profilom

A profilom.php fájl a felhasználói profil kezelését biztosítja, lehetővé téve az adatok megtekintését és szerkesztését.

Főbb Funkciók:

- Adatok betöltése a felhasználó táblából.
- Interaktív szerkesztés ceruza ikonnal.
- Kliensoldali validáció és megerősítő modal.

## 8.7 Rendelés

A rendeles.php fájl a rendelési folyamat véglegesítését kezeli, szállítási adatokkal és e-mail visszaigazolással.

Főbb Funkciók:

- Kosár tartalmának áttekintése.
- Szállítási adatok validálása.
- Email visszaigazolás PHPMailer-rel.
- Sikerüzenet modal ablakkal.

## 8.8 Rendelés Megtekintése

A rendeles\_megtekint.php fájl a korábbi rendelések áttekintését biztosítja rezponzív kártyákkal.

Főbb Funkciók:

- Rendelések lekérdezése és csoportosítása.
- Részletes adatok megjelenítése (azonosító, dátum, állapot, tételek).
- Bejelentkezés ellenőrzése.

## 8.9 Visszajelzések


**KÜLDD EL A VÉLEMÉNYED!**

MENNYIRE ELÉGEDETT? (1-5 CSILLAG)

★★★★★

VISSZAJELZÉS SZÖVEGE

Írd meg a véleményed...

☐ Nem vagyok robot  reCAPTCHA  
Adatvédelem - Általános Szolgáltatás Feltételek

**KÜLDÉS**

A véleményküldő felület célja, hogy a felhasználók könnyedén visszajelzést adhassanak az általuk tapasztalt élményekről. A felület tetején egy figyelemfelkeltő cím szerepel: „Küldd el a véleményed!”, amely ösztönzi az interakciót. Az értékelés egy ötfokozatú csillagrendszer segítségével történik, ahol a felhasználó megjelölheti, mennyire volt elégedett az adott szolgáltatással vagy termékkel. Ezt követően egy szöveges mező áll rendelkezésre, ahová a részletes visszajelzés írható, például pozitívumok, javaslatok vagy negatív tapasztalatok.

A biztonság és az automatizált beküldések elkerülése érdekében a felület reCAPTCHA ellenőrzéssel van ellátva, amely igazolja, hogy valódi felhasználó küldi el a véleményt. Végül egy jól látható, narancssárga „Küldés” gomb biztosítja az adat elküldését. Az egész rendszer letisztult, felhasználóbarát megjelenésű, és könnyen kezelhető bármilyen eszközön.

## 9 Kosárkezelő Rendszer Dokumentáció

### 9.1 Bevezetés

A kosárkezelő rendszer négy PHP szkripttel (`kosarba_rakas.php`, `kosarbol_torles.php`, `kosar_uritesi.php`, `kosar_mod.php`) biztosítja a kosárfunkciókat bejelentkezett és vendég felhasználók számára. A bejelentkezett felhasználók adatai adatbázisban, a vendég felhasználók adatai session-ben és sütiben tárolódnak.

### 9.2 Rendszeráttekintés

A rendszer lehetővé teszi a tételek hozzáadását, törlését, mennyiség módosítását és a kosár ürítését. A szkriptek JSON bemenetet fogadnak és JSON válaszokat küldenek.



## 9.3 Adatbázis-séma

- tetelek tábla:
  - felhasználó\_id (integer): Felhasználó azonosítója.
  - etel\_id (integer): Tétel azonosítója.
  - darab (integer): Mennyiség.

## 9.4 Munkamenet- és Sütikezelés

- Munkamenet: `$_SESSION['felhasznalo_id']s$_SESSION['kosar']`. Süti : `guest_cartJSONkdolssti,7`

## 9.5 Szkriptek Részletes Bemutatása

### 9.5.1 kosarba\_rakas.php

Funkció: Tétel hozzáadása vagy mennyiség növelése.

- Bejelentkezett: Tétel keresése, frissítés vagy beszúrás a tetelek táblába.
- Vendég: `$_SESSION['kosar']mdostsa,guest_cartstifrisstse`.

```

1  <?php
2  session_start();
3  include 'adatbazisra_csatlakozas.php';
4
5  $data = json_decode(file_get_contents('php://input'), true);
6  $itemId = $data['itemId'];
7
8  // After successful login authentication
9  if (isset($_SESSION['felhasznalo_id'])) {
10     // Logged in user - update database
11     $userId = $_SESSION['felhasznalo_id'];
12
13     // Check if the item already exists in the cart
14     $query = "SELECT darab FROM tetelek WHERE felhasznalo_id = ? AND etel_id = ?";
15     $stmt = $conn->prepare($query);
16     $stmt->bind_param("ii", $userId, $itemId);
17     $stmt->execute();
18     $result = $stmt->get_result();
19
20     if ($result->num_rows > 0) {
21         // If the item exists, increment the quantity
22         $row = $result->fetch_assoc();
23         $newQuantity = $row['darab'] + 1;
24
25         $updateQuery = "UPDATE tetelek SET darab = ? WHERE felhasznalo_id = ? AND etel_id = ?";
26         $updateStmt = $conn->prepare($updateQuery);
27         $updateStmt->bind_param("iii", $newQuantity, $userId, $itemId);
28         $updateStmt->execute();
29     } else {
30         // If the item doesn't exist, insert it with a quantity of 1
31         $insertQuery = "INSERT INTO tetelek (felhasznalo_id, etel_id, darab) VALUES (?, ?, 1)";
32         $insertStmt = $conn->prepare($insertQuery);
33         $insertStmt->bind_param("ii", $userId, $itemId);
34         $insertStmt->execute();
35     }
36 } else {
37     // Guest user - update session and cookie
38     if (!isset($_SESSION['kosar'])) {
39         $_SESSION['kosar'] = [];
40     }
41     $_SESSION['kosar'][$itemId] = isset($_SESSION['kosar'][$itemId]) ? $_SESSION['kosar'][$itemId] + 1 : 1;
42
43     // Update cookie with 7-day expiration
44     setcookie('guest_cart', json_encode($_SESSION['kosar']), time() + 604800, '/');
45 }
46
47 echo json_encode(['success' => true]);
48 >>

```

### 9.5.2 kosarbol\_torles.php

Funkció: Tétel törlése a kosárból.

Működés:

- Bejelentkezett: Tétel törlése a tetelek táblából.
- Vendég: Tétel eltávolítása  $$_SESSION['kosar']$  – *bl,stifrisstse.*

```

1  <?php
2  session_start();
3  include 'adatbazisra_csatlakozas.php';
4
5  $data = json_decode(file_get_contents('php://input'), true);
6  $itemId = $data['itemId'];
7
8  if (isset($_SESSION['felhasznalo_id'])) {
9      // Logged in user - delete from database
10     $userId = $_SESSION['felhasznalo_id'];
11     $query = "DELETE FROM tetelek WHERE felhasznalo_id = ? AND etel_id = ?";
12     $stmt = $conn->prepare($query);
13     $stmt->bind_param("ii", $userId, $itemId);
14     $stmt->execute();
15 } else {
16     // Guest user - delete from session
17     if (isset($_SESSION['kosar'][$itemId])) {
18         unset($_SESSION['kosar'][$itemId]);
19         // Update cookie
20         setcookie('guest_cart', json_encode($_SESSION['kosar']), time() + 604800, '/');
21     }
22 }
23
24 echo json_encode(['success' => true]);
25 ?>

```

### 9.5.3 kosar\_uritese.php

Funkció: Kosár teljes ürítése.

- Bejelentkezett: Összes tétel törlése a tetelek táblából.
- Vendég:  $$_SESSION['kosar']$ sguest\_carttrlse.

```

1  <?php
2  session_start();
3  include 'adatbazisra_csatlakozas.php';
4
5  $response = ['success' => false];
6
7  if (isset($_SESSION['felhasznalo_id'])) {
8      // Bejelentkezett felhasználó - töröljük az adatbázisból
9      $userId = $_SESSION['felhasznalo_id'];
10     $deleteQuery = "DELETE FROM tetelek WHERE felhasznalo_id = ?";
11     $deleteStmt = $conn->prepare($deleteQuery);
12     $deleteStmt->bind_param("i", $userId);
13
14     if ($deleteStmt->execute()) {
15         $response['success'] = true;
16     }
17 } else {
18     // Vendég felhasználó - töröljük a session-ből és a cookie-ból
19     unset($_SESSION['kosar']);
20     setcookie('guest_cart', '', time() - 3600, '/');
21     $response['success'] = true;
22 }
23
24 // JSON válasz küldése
25 header('Content-Type: application/json');
26 echo json_encode($response);
27 exit();
28 >
29

```

### 9.5.4 kosar\_mod.php

Funkció: Tétel mennyiségének növelése vagy csökkentése.

```

<?php
session_start();
include 'adatbazisra_csatlakozas.php';

$data = json_decode(file_get_contents('php://input'), true);
$itemId = $data['itemId'];
$action = $data['action'];
$response = ['success' => false, 'newQuantity' => 0];

try {
    if (isset($_SESSION['felhasznalo_id'])) {
        // Logged in user - update database
        $userId = $_SESSION['felhasznalo_id'];

        // Check if the item exists in the cart
        $stmt = $conn->prepare("SELECT darab FROM tetelek WHERE felhasznalo_id = ? AND etel_id = ?");
        $stmt->bind_param("ii", $userId, $itemId);
        $stmt->execute();
        $result = $stmt->get_result();

        if ($result->num_rows > 0) {
            // If the item exists, update the quantity
            $row = $result->fetch_assoc();
            $newQuantity = $row['darab'];

            if ($action === "increase") {
                $newQuantity++;
            } else {
                $newQuantity--;
            }

            // If the quantity is 0 or less, delete the item
            if ($newQuantity <= 0) {
                $deleteQuery = "DELETE FROM tetelek WHERE felhasznalo_id = ? AND etel_id = ?";
                $deleteStmt = $conn->prepare($deleteQuery);
                $deleteStmt->bind_param("ii", $userId, $itemId);
                $deleteStmt->execute();
                $newQuantity = 0;
            } else {
                // Otherwise, update the quantity
                $updateQuery = "UPDATE tetelek SET darab = ? WHERE felhasznalo_id = ? AND etel_id = ?";
                $updateStmt = $conn->prepare($updateQuery);
                $updateStmt->bind_param("iii", $newQuantity, $userId, $itemId);
                $updateStmt->execute();
            }
        }
    }
}

```

```

    } else {
        // Guest user - update session
        if (!isset($_SESSION['kosar'])) {
            $_SESSION['kosar'] = [];
        }

        if (isset($_SESSION['kosar'][$itemId])) {
            if ($action === "increase") {
                $_SESSION['kosar'][$itemId]++;
            } else {
                $_SESSION['kosar'][$itemId]--;
            }

            $newQuantity = $_SESSION['kosar'][$itemId];

            // If the quantity is 0 or less, remove the item
            if ($_SESSION['kosar'][$itemId] <= 0) {
                unset($_SESSION['kosar'][$itemId]);
                $newQuantity = 0;
            }
        }
    }

    $response = ['success' => true, 'newQuantity' => $newQuantity];
} catch (Exception $e) {
    $response['error'] = $e->getMessage();
}

echo json_encode($response);
?>

```

- Bejelentkezett: Mennyiség módosítása vagy tétel törlése a tetelek táblában.
- Vendég: `$_SESSION['kosar']` módosítása, tisztítása.

## 9.6 Biztonsági Szempontok

- SQL injekció elleni védelem előkészített lekérdezésekkel.
- Munkamenet- és sütikezelés biztonságos beállítása.

## 9.7 Összegzés

A kosárkezelő rendszer hatékony és biztonságos megoldást nyújt a kosárfunkciók kezelésére, JSON-alapú kommunikációval és modularitással.

# 10 Használt technológiák

A projekt során többféle technológiát és platformot használtunk, amelyek különböző célokat szolgáltak: a programozási környezet kialakításától kezdve a kommunikáción és verziókezelésen át egészen a kreatív tartalomgyártásig. Az alábbiakban részletesen bemutatjuk ezeket az eszközöket és technológiákat.

## 1. Visual Studio

**Szerepe:** Programozási környezet

**Mikor használtuk:** A projekt teljes fejlesztési szakaszában

**Miért használtuk?**

A Visual Studio-t választottuk fő fejlesztőkörnyezetként, mivel rendkívül hatékony, stabil és támogatja az általunk használt programozási nyelveket (például C#, Python, HTML/CSS/JS vagy más projektfüggő technológiák).

Lehetőséget nyújt a hibakeresésre (debug), a kód formázására, valamint integrálható más eszközökkel, például a GitHubbal.

---

## 2. Gmail Chat

**Szerepe:** Kommunikációs eszköz

**Mikor használtuk:** A projekt tervezési és haladási egyeztetéseinél

**Miért használtuk?**

A Gmail Chat-et az egymás közötti gyors, írásos kommunikációra használtuk, főként akkor, amikor rövid, céltudatos egyeztetésekre volt szükség. Előnye, hogy minden üzenet visszakereshető, és könnyen lehet benne csatolmányokat, linkeket küldeni, akár mobilon, akár gépen keresztül.

---

## 3. Discord

**Szerepe:** Kommunikáció, élő megbeszélések, képernyőmegosztás

**Mikor használtuk:** Fejlesztési időszak alatt rendszeresen

**Miért használtuk?**

A Discord-ot élő hangalapú megbeszélésekhez és képernyőmegosztáshoz használtuk, különösen a közös munkavégzés során. Nagy előnye volt, hogy gyorsan tudtunk rajta hívásokat indítani, és könnyen mutathattuk egymásnak a kódot, tervrajzokat vagy hibákat valós időben.

---

## 4. GitHub

**Szerepe:** Verziókezelés, csapatmunka

**Mikor használtuk:** A fejlesztés megkezdésétől egészen a projekt lezárásáig

**Miért használtuk?**

A GitHub-on tároltuk a projekt kódját, és itt kezeltük a verziókat, branch-eket, valamint itt végeztük el az összevonásokat (merge).

Segítségével egyszerre több fejlesztő is tudott dolgozni ugyanazon a projekten anélkül, hogy felülírtuk volna egymás munkáját. A commit üzenetek révén jól követhetővé vált a haladás.

---

## 5. AI alapú képgeneráló eszközök

**Szerepe:** Kreatív tartalom (pl. képek, illusztrációk) létrehozása

**Mikor használtuk:** A projekt látványvilágának kialakításakor

**Miért használtuk?**

A mesterséges intelligenciát alkalmazó képgeneráló eszközökkel (Freepik) egyedi, vizuálisan látványos képeket tudunk előállítani a projekt különböző elemeihez (pl. háttérképek, karakterek, grafikai elemek). Ez lehetővé tette, hogy olyan minőségi vizuális tartalmakat hozzunk létre, amelyek saját készítésűnek számítanak, és egyedivé tették a végeredményt.

## Tesztelés

A weboldal projekt fejlesztése során a minőségbiztosítás érdekében alapos tesztelést végeztünk a frontend és backend komponenseken egyaránt. A tesztelés célja a rendszer megbízhatóságának, helyes működésének és a felhasználói élmény magas színvonalának garantálása volt. A tesztelési folyamat három fő területre fókuszált: a HTML/CSS frontend, a PHP backend és a JavaScript kliensoldali logika. Az alábbiakban bemutatjuk a felhasznált eszközöket és a tesztelési megközelítéseket.

### Frontend tesztelés (HTML/CSS)

A felhasználói felület helyes megjelenésének és funkcionalitásának ellenőrzésére **Cypress** end-to-end tesztelési keretrendszert használtunk. A Cypress lehetővé tette valós böngészőkörnyezetben való tesztelést, így biztosítva, hogy a HTML struktúra és a CSS stílusok konzisztensen működjenek különböző böngészőkben (pl. Chrome, Firefox, Edge). A tesztek során ellenőriztük:

- Az oldalak reszponzivitását különböző képernyőméreteken.
- A CSS stílusok helyes alkalmazását (pl. elrendezés, színek, animációk).
- Az interaktív elemek (pl. gombok, űrlapok) megfelelő működését. A Cypress tesztek automatizáltan futottak a CI/CD pipeline-ban, így a kódváltozások gyorsan validálhatók voltak.

### Backend tesztelés (PHP)

A szerveroldali logika tesztelésére **PHPUnit** keretrendszert alkalmaztunk, amely a PHP kód egység- és integrációs tesztelésére szolgál. A PHPUnit segítségével a következőket vizsgáltuk:

- Az üzleti logika helyes működését (pl. adatfeldolgozás, számítások).
- Az adatbázis-műveletek pontosságát (pl. CRUD műveletek).
- A hibakezelés és kivételkezelés hatékonyságát. A tesztek különálló környezetben futottak, mockolt adatbázisokkal, hogy elkerüljük a valós adatok módosítását. A kódlefedettségi riportok alapján biztosítottuk, hogy a kritikus funkciók alaposan le legyenek tesztelve.

### JavaScript tesztelés

A kliensoldali JavaScript kód tesztelésére **Jest** keretrendszert használtunk. A Jest gyors és hatékony egységtesztelést tett lehetővé, amelynek során a következőket ellenőriztük:

- A JavaScript függvények helyes működését (pl. eseménykezelők, adatmanipuláció).
- A DOM-műveletek pontosságát.

- Az aszinkron műveletek (pl. API-hívások) megfelelő kezelését. A Jest snapshot tesztelési funkcióival validáltuk a dinamikusan generált tartalmak konzisztenciáját. A tesztek mockolt környezetben futottak, hogy izoláljuk a külső függőségeket.

## Összegzés

A tesztelési folyamat során a Cypress, PHPUnit és Jest eszközök kombinációjával átfogó ellenőrzést végeztünk a weboldal minden rétegén. A tesztek automatizált futtatása és a kódlefedettségi riportok elemzése biztosította, hogy a projekt stabil és megbízható legyen. A hibák korai felderítése és javítása jelentősen hozzájárult a fejlesztési ciklus hatékonyságához és a végtermék minőségéhez. A tesztelési eredmények dokumentálva vannak, és a jövőbeli fejlesztések során tovább bővíthetők.

## 11 Következtetés

A FlavorWave webalkalmazás egy robusztus, biztonságos és felhasználóbarát platformot kínál online ételrendeléshez. A jól strukturált adatbázis, reszponzív felületek és átfogó funkciók kielégítik mind az ügyfelek, mind a személyzet igényeit. A fejlesztési idővonal és feladatleosztás biztosítja az időbeni átadást, szigorú teszteléssel és optimalizálással alátámasztva.