

SZABADKAI MŰSZAKI SZAKFŐISKOLA

PROJEKTUM

Web programozás tantárgyból

HALLGATÓK

Oravec Anett
Basic Palkovic David

MENTOR

prof. dr Zlatko Čović

SZABADKA, 2024.

Tartalom

1. A feladatkör meghatározása	2
2. Kidolgozás	3
3. Az adatbázis struktúrája (MySQL Workbench)	10
4. A projekt működésének leírása	12
5. Felhasznált irodalom	20

1. A feladatkör meghatározása

A projekt webcíme : <https://teammates.stud.vts.su.ac.rs/proji/register/>

A projekt célja egy webalkalmazás fejlesztése az ételek egyszerű és hatékony szállításához egy étteremből. Az alkalmazás négyféle hozzáférési szintet különböztet meg: vendég, regisztrált felhasználó, szállító és rendszergazda.

A vendég megtekintheti az oldalon kínált információkat, értékeléseket és megjegyzéseket, valamint regisztrálhat az oldalon. A regisztrált felhasználó ételt rendelhet, kezelheti profiladatait, ellenőrizheti megrendeléseit és kezelheti kedvenc ételeit. A szállító rögzítheti a szállítás állapotát, és megjegyzéseket fűzhet a szállításhoz. A rendszergazda létrehozhat, módosíthat és törölhet étel-kategóriákat és ételeket, kezelheti a szállítók adatait, és hozzárendelhet rendeléseket szállítókhoz. A felhasználói regisztrációhoz biztonságos e-mail aktiválás szükséges, és nem engedélyezett azonos e-mail címmel több fiók regisztrálása. A bejelentkezett felhasználók kategóriák alapján tekinthetik meg az ételeket, és többféle ételt is rendelhetnek egy rendelésen belül.

A megrendelés előtt meg kell adni a szállítási címet, és a sikeres megrendelésről e-mail értesítést kapnak. Az értékelés és a megjegyzések csak egyszer végezhetők el egy rendelésre, és nem módosíthatók. Az oldal adminisztrációját PHP munkamenetekkel kell védeni, és a jelszavakat bcrypt algoritmussal kell hash-elni. Az adatbázis food nevű lesz, és tartalmazza az összes szükséges táblázatot a projekt funkcionalitásának kielégítésére.

2. Kidolgozás

Regisztrált / bejelentkezett felhasználó rendelhet ételt az étteremből, ami így néz ki:

```
try {
    // Calculate total price of the order
    $totalPrice = 0;

    foreach ($quantities as $foodId => $quantity) {
        // Get the food price and discount from the database
        $sql = "SELECT food_price, food_discount FROM `Food`
WHERE food_id = :food_id";
        $stmt = $conn->prepare($sql);
        $stmt->execute(['food_id' => $foodId]);
        $food = $stmt->fetch(PDO::FETCH_ASSOC);
        $foodPrice = $food['food_price'];
        $foodDiscount = $food['food_discount'];

        // Calculate the actual price considering discount
        if ($foodDiscount !== null) {
            // If food_discount is not null, calculate price with
discount
            $actualPrice = $foodDiscount * $quantity;
        } else {
            // If food_discount is null, use the original
food_price
            $actualPrice = $foodPrice * $quantity;
        }

        // Increase the total order price by the current item's
cost multiplied by quantity
        $totalPrice += $actualPrice;
    }

    // Insert the order into the Orders table
    $sql = "INSERT INTO `Orders` (id_user, order_date, order_city,
order_street, order_status, order_price, phone)
VALUES (:id_user, NOW(), :order_city, :order_street,
0, :order_price, :phone)";
    $stmt = $conn->prepare($sql);
```

```

$stmt->execute([
    'id_user' => $userId,
    'order_city' => $city,
    'order_street' => $street,
    'order_price' => $totalPrice, // Use the calculated total
price here
    'phone' => $phone
]);
$orderId = $conn->lastInsertId();

// Insert ordered food items into OrderedFood table
foreach ($quantities as $foodId => $quantity) {
    // Insert the ordered food items into the OrderedFood
table
    $sql = "INSERT INTO `OrderedFood` (order_id, food_id,
order_amount) VALUES (:order_id, :food_id, :order_amount)";
    $stmt = $conn->prepare($sql);
    $stmt->execute(['order_id' => $orderId, 'food_id' =>
$foodId, 'order_amount' => $quantity]);
}

// Commit the transaction
$conn->commit();

// Send an order confirmation email
sendOrderConfirmationEmail($userId, $orderId, $conn);

// Store the success message in the session
$_SESSION['message'] = "Rendelés sikeresen leadva! Az e-mail
sikeresen elküldve.";
} catch (Exception $e) {
    // Roll back the transaction in case of error
    $conn->rollBack();
    // Store the error message in the session
    $_SESSION['message'] = "Hiba történt a rendelés feldolgozása
során.";
}
// Redirect to food.php
header("Location: food.php");
exit();
} else {

```

Regisztrált / bejelentkezett felhasználó ellenőrizheti megrendeléseit, ami így néz ki:

```
<?php if (!empty($orders)): ?>
    <div class="container mt-5">
        <div class="row">
            <div class="col-md-6 mx-auto">
                <div class="card">
                    <div class="card-header">
                        Aktuális rendelések
                    </div>
                    <div class="card-body">
                        <?php foreach ($orders as $order): ?>
                            <div class="card mb-3">
                                <div class="card-body">
                                    <p><strong>Rendelési
dátum:</strong> <?= htmlspecialchars($order['order_date']) ?></p>
                                    <p><strong>Város:</strong> <?=
htmlspecialchars($order['order_city']) ?></p>
                                    <p><strong>Utca:</strong> <?=
htmlspecialchars($order['order_street']) ?></p>
                                    <p><strong>Ár:</strong> <?=
htmlspecialchars($order['order_price']) ?> RSD</p>
                                </div>
                            </div>
                        <?php endforeach; ?>
                    </div>
                </div>
            </div>
        </div>
    </div>
<?php endif; ?>
```

Rendszergazda odarendelhet rendelést egy szállítóhoz ami így néz ki:

PHP kód:

```
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    if (isset($_POST['assign_delivery'])) {
        $order_id = $_POST['order_id'];
        $del_id = $_POST['delivery_person'];

        try {
            $stmt = $conn->prepare("UPDATE Orders SET del_id
= :del_id, admin_id = :admin_id WHERE order_id = :order_id");
            $stmt->bindParam(':del_id', $del_id);
            $stmt->bindParam(':admin_id', $admin_id);
            $stmt->bindParam(':order_id', $order_id);
            $stmt->execute();
        } catch (PDOException $e) {
            die("Failed to update order: " . $e->getMessage());
        }
    }
}
```

HTML kód:

```
<form action="" method="post">
    <input type="hidden" name="order_id"
value="<?php echo $order['order_id']; ?>">
    <select name="delivery_person"
class="form-select">
        <?php foreach ($deliveryPersons as
$person): ?>
            <option value="<?php echo
$person['del_id']; ?>"><?php echo $person['del_email']; ?></option>
        <?php endforeach; ?>
    </select>
</td>
<td>
        <button type="submit"
name="assign_delivery" class=" btn-custom">Hozzárendel</button>
</form>
```

Létre kellett hozni egy Keresés opciót, amelyen keresztül a regisztrált felhasználó kulcsszavak alapján keresést végez az ételkínálatban ami így néz ki:

PHP kód:

```
// Keresési logika
$searchTerm = isset($_POST['searchTerm']) ? $_POST['searchTerm'] : '';
$category = isset($_POST['category']) ? $_POST['category'] : '';

if ($searchTerm && !$category) {
    // Search by food name only
    $sql = "SELECT * FROM `Food` WHERE `food_name` LIKE :searchTerm";
    $stmt = $conn->prepare($sql);
    $stmt->execute(['searchTerm' => "%$searchTerm%"]);
} elseif (!$searchTerm && $category) {
    // Filter by category only
    $sql = "SELECT * FROM `Food` WHERE `food_category` = :category";
    $stmt = $conn->prepare($sql);
    $stmt->execute(['category' => $category]);
} elseif ($searchTerm && $category) {
    // Search by food name and filter by category
    $sql = "SELECT * FROM `Food` WHERE `food_name` LIKE :searchTerm
AND `food_category` = :category";
    $stmt = $conn->prepare($sql);
    $stmt->execute(['searchTerm' => "%$searchTerm%", 'category' =>
$category]);
} else {
    // No search or category filter, fetch all
    $sql = "SELECT * FROM `Food`";
    $stmt = $conn->prepare($sql);
    $stmt->execute();
}
```

HTML kód:

```
<form method="POST" class="mb-4">
    <div class="input-group mb-3">
        <input type="text" class="form-control" name="searchTerm"
placeholder="Keresés étel neve alapján">
        <button type="submit" class="btn btn-
primary">Keresés</button>
    </div>
```



```

        <div class="input-group mb-3">
            <label class="input-group-text"
for="categorySelect">Kategória:</label>
            <select class="form-select" id="categorySelect"
name="category">
                <option value="">Összes kategória</option>
                <?php
// Fetch all distinct food categories from the
database
                $sql_categories = "SELECT DISTINCT food_category FROM
`Food`";
                $stmt_categories = $conn->prepare($sql_categories);
                $stmt_categories->execute();
                $categories = $stmt_categories-
>fetchAll(PDO::FETCH_COLUMN);

                // Output each category as an option in the select
dropdown
                foreach ($categories as $category) {
                    echo "<option
value='{ $category }'>{ $category }</option>";
                }
                ?>
            </select>
        </div>
    </form>

```

Mentorunk kérésére létrehoztuk egy kétfaktoros hitelesítést amit az admin felület belépésére használtunk, ami így néz ki:

```

function generateSecret() {
    return mt_rand(100000, 999999);
}

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    if (isset($_POST['verify_secret'])) {
        $entered_secret = $_POST['secret'];
        if (isset($_SESSION['admin_email'])) {
            $stmt = $conn->prepare("SELECT secret FROM Admin WHERE
admin_email = :email");
            $stmt->execute(['email' => $_SESSION['admin_email']]);
            $admin = $stmt->fetch();

```

```

        if ($admin && $entered_secret == $admin['secret']) {
            // Clear the secret in the database
            $stmt = $conn->prepare("UPDATE Admin SET secret =
NULL WHERE admin_email = :email");
            $stmt->execute(['email' => $_SESSION['admin_email']]);

            // Redirect to menu.php
            header("Location: menu.php");
            exit();
        } else {
            $error = "Invalid secret code.";
        }
    } else {
        $error = "Session expired. Please log in again.";
    }
} else {
    $email = trim($_POST['email']);
    $password = $_POST['password'];

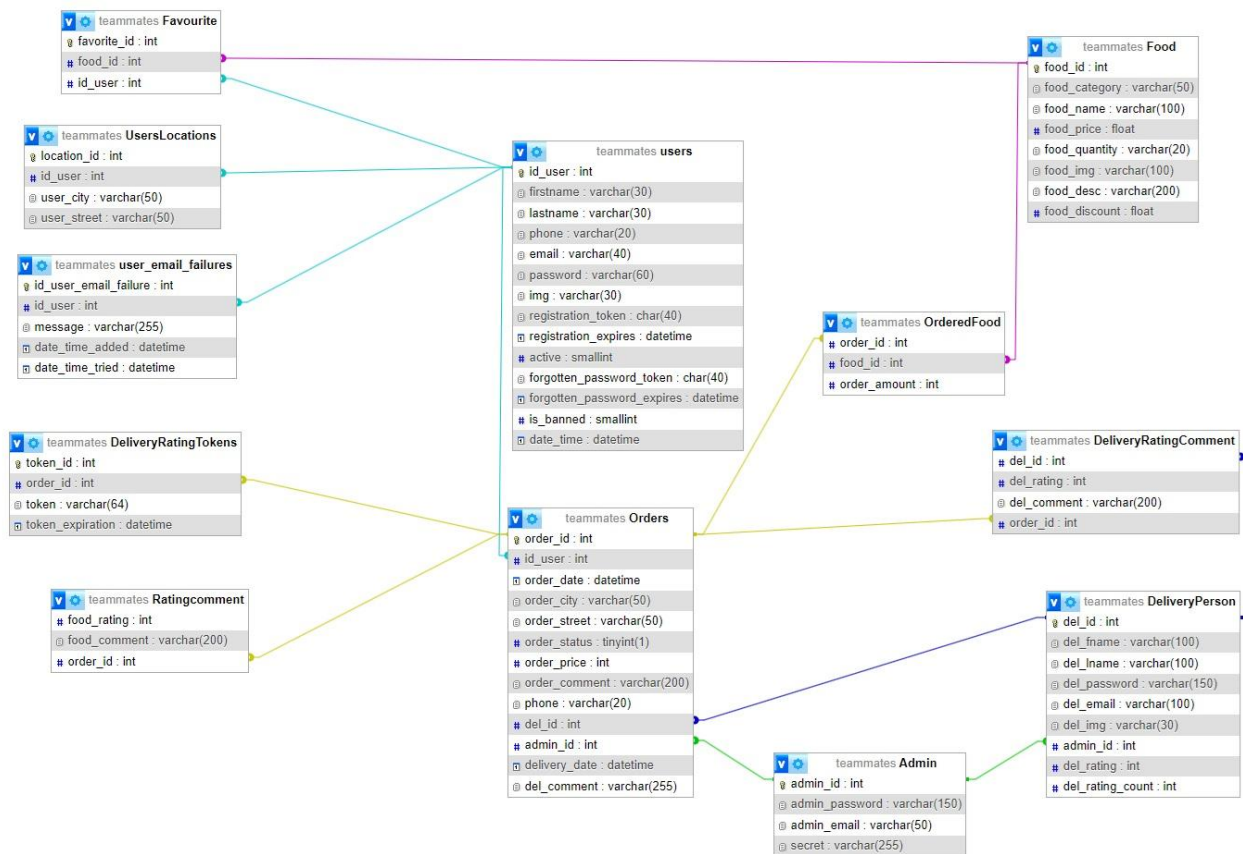
    // Validate input
    if (!empty($email) && !empty($password)) {
        // Prepare and execute the query
        $stmt = $conn->prepare("SELECT * FROM Admin WHERE
admin_email = :email");
        $stmt->execute(['email' => $email]);
        $admin = $stmt->fetch();

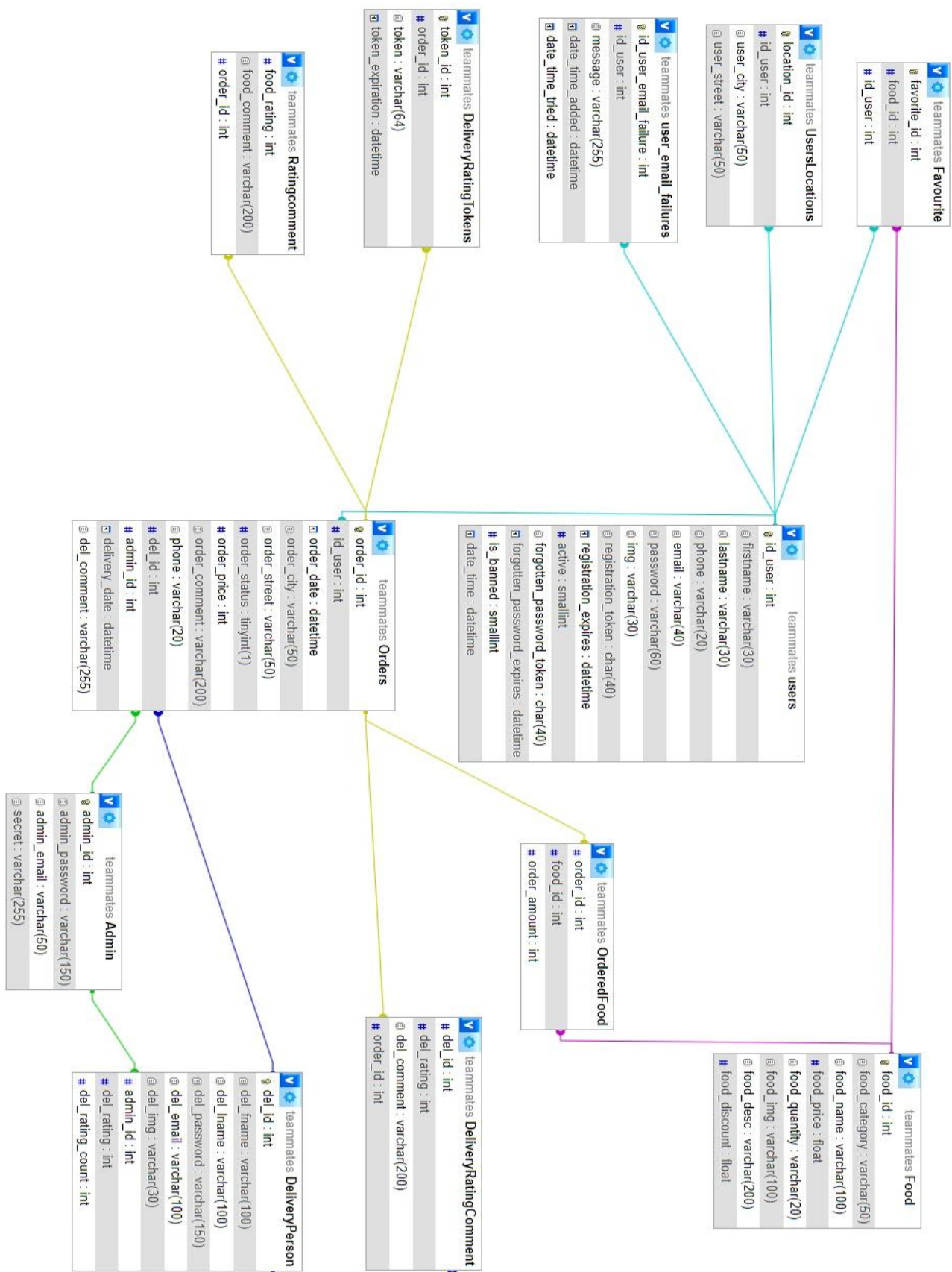
        // Verify hashed password
        if ($admin && password_verify($password,
$admin['admin_password'])) {
            // Set session variables
            $_SESSION['admin_email'] =
trim($admin['admin_email']);

            // Generate and store secret in database
            $secret = generateSecret();
            $stmt = $conn->prepare("UPDATE Admin SET secret
= :secret WHERE admin_email = :email");
            $stmt->execute(['secret' => $secret, 'email' =>
$_SESSION['admin_email']]);

```

3. Az adatbázis struktúrája (MySQL Workbench)

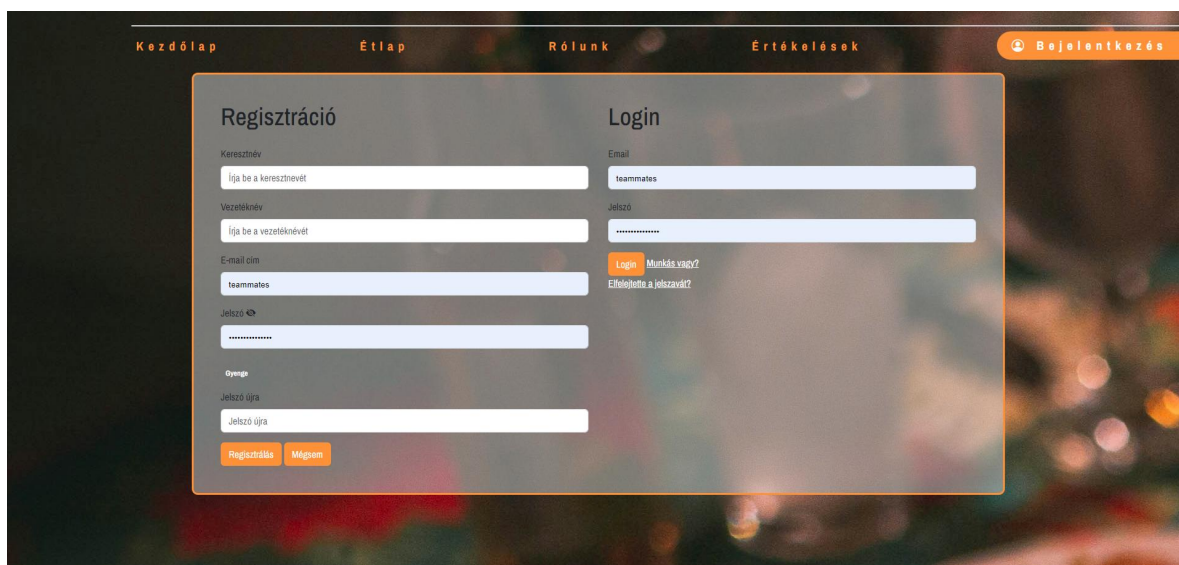




4. A projekt működésének leírása

A weboldalon elérhetők a regisztráció és a bejelentkezés funkciók. A felhasználók a regisztrációs oldalon adhatják meg adataikat, ahol minden mezőt ki kell tölteniük. A jelszónak minimum 8 karakter hosszúnak kell lennie, tartalmaznia kell nagybetűt és speciális karaktert is. Sikeres regisztráció után a felhasználó kap egy e-mailt, amelyben található aktiválási linket el kell fogadnia. Az aktiválás után a felhasználó már bejelentkezhet az oldalra.

A "Munkás Vagy?" részlegnél a szállítók jelentkezhetnek be. A bejelentkezéshez meg kell adniuk a felhasználónevüket és jelszavukat. A rendszer ellenőrzi az adatokat, és sikeres hitelesítés esetén hozzáférést biztosít a megfelelő funkciókhoz.



The screenshot displays the Teammates website interface. At the top, there is a navigation bar with links: **Kezdőlap**, **Étlap**, **Rólunk**, **Értékelések**, and a **Bejelentkezés** button with a user icon. The main content area is divided into two columns: **Regisztráció** (Registration) and **Login**.

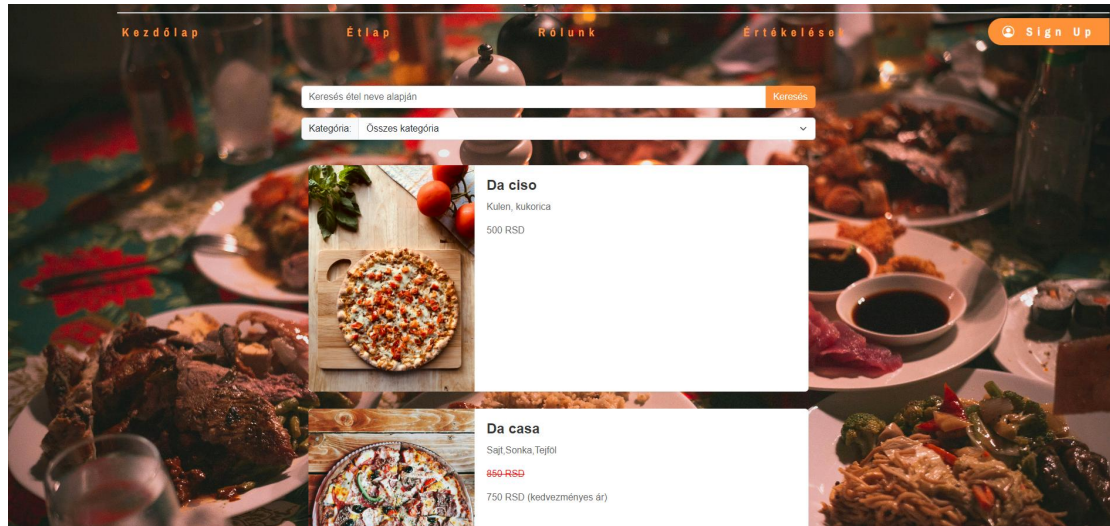
Regisztráció (Registration) form:

- Keresztnév:** Input field with placeholder "Írja be a keresztnévét".
- Vezetéknév:** Input field with placeholder "Írja be a vezetéknévét".
- E-mail cím:** Input field with placeholder "teammates".
- Jelszó:** Input field with a strength indicator icon.
- Újra:** Input field with placeholder "Jelszó újra".
- Buttons:** **Regisztráció** and **Mégsem**.

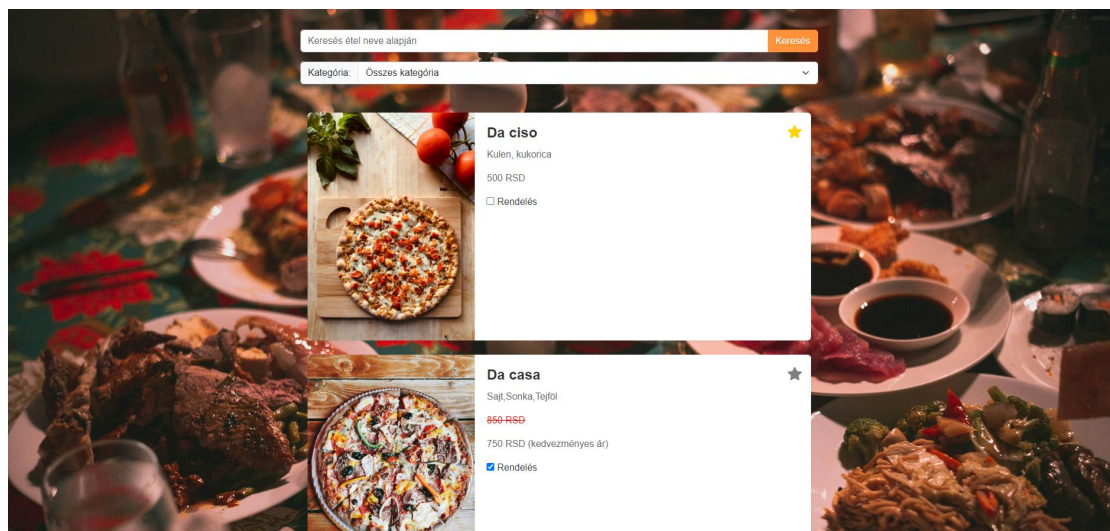
Login form:

- Email:** Input field with placeholder "teammates".
- Jelszó:** Input field with placeholder ".....".
- Buttons:** **Login** and **Munkás vagy?**.
- Link:** **Elfelejtette a jelszavát?**.

A nem bejelentkezett felhasználó az étlap menüpontnál megtekintheti az ételeket és láthatja, mely ételek vannak akcióban. Ha részletesebben szeretne böngészni az ételek között, regisztrálnia kell, vagy be kell jelentkeznie. A regisztráció után a felhasználók további funkciókat is elérhetnek, mint például az ételrendelés, a profiladatok módosítása, a kedvenc ételek listájának kezelése és a megrendelések nyomon követése.



A bejelentkezett felhasználó :



Az ételrendelésnél a felhasználók módosíthatják az étel mennyiségét, valamint megváltoztathatják a telefonszámot, a várost és az utca adatokat is, ha nem a megadott címre szeretnék rendelni. A rendelés gomb megnyomása után a felhasználó e-mailt kap, amelyben megerősítjük a rendelését.

Rendelés részletei

Étel	Leírás	Ár (RSD)	Mennyiség	Összeg (RSD)
Da casa	Sajt, Sonka, Tejföl	750.00	2	750.00
				Végösszeg (RSD): 1500.00

Szállítási adatok

Telefonszám
0600142949

Város
Subotica

Utica
Matija

Rendelés

A profil menüpont alatt a felhasználó módosíthatja az adatait, megtekintheti kedvenc ételeit és az aktuális rendelését is.

Felhasználói információk

Név: David Basic
Email: basicdavid79@gmail.com
Telefonszám: 0600142949
Város: Subotica
Utica: Matija
Profil szerkesztése

Kedvenc ételek

Da ciso - 500 RSD
Rendelés

Aktuális rendelések

Rendelési dátum: 2024-06-23 21:43:30
Város: Subotica
Utica: Matija
Ár: 900 RSD

Profil adatok szerkeztése :

Profil szerkesztése

Keresztnév

David

Vezetéknév

Basic

Telefonszám

0600142949

Város

Subotica

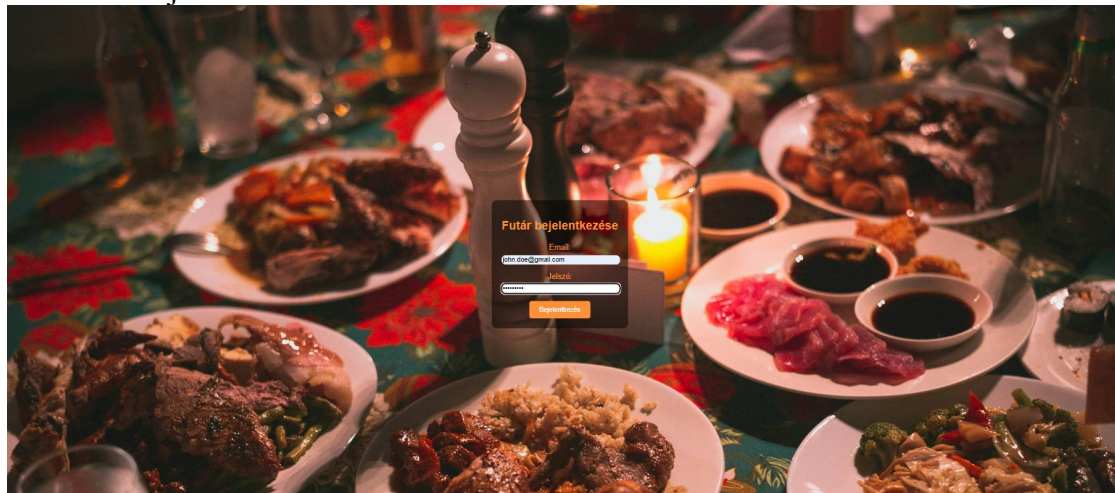
Utca

Matija

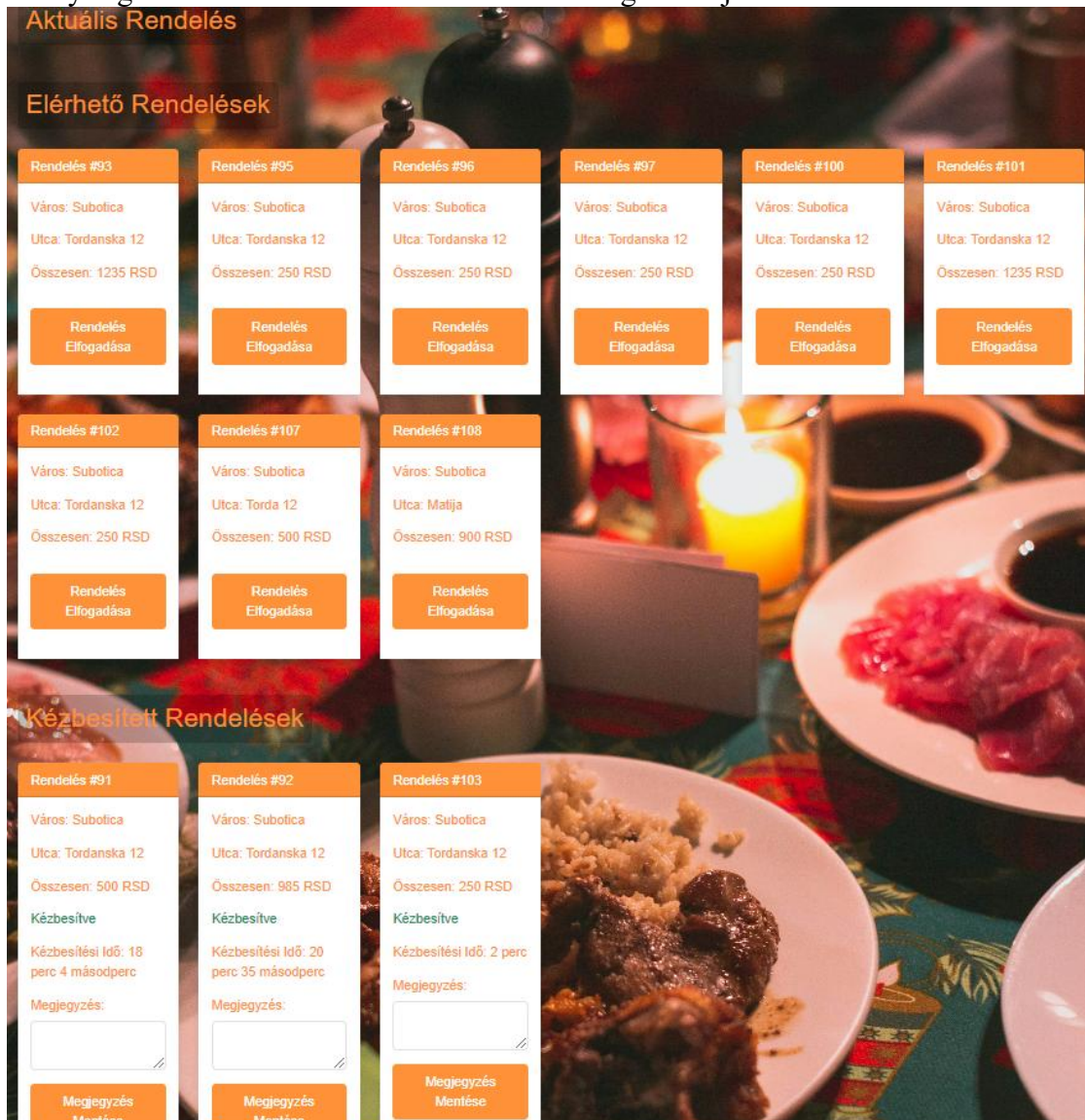
Bezárás

Változások mentése

A futárok bejelentkezése :



Miután bejelentkezett, a futár láthatja az aktuális rendeléseket és a kézbesített rendeléseket 24 óráig. Emellett megjegyzést is írhat a kézbesített rendelésekhez, amely segíthet a későbbi referenciákban és a szolgáltatás javításában:



Amikor a futár elfogadja a rendelést, az az "Aktuális rendelés" szekcióba kerül, és csak ezt a rendelést fogja látni. Emellett megtekintheti az előzőleg ugyanahhoz a címhez írt megjegyzéseket a kollégáitól, amelyek segíthetnek a pontos és hatékony kézbesítésben. A felhasználó kap egy emailt ezek után ahol bírja értékelni az ételt és a kézbesítőt is.

Aktuális Rendelés

Rendelés #93

Város: Subotica

Ulica: Tordanska 12

Összesen: 1235 RSD

Kézbesítő megjegyzések:

Sok kutya
Undok volt az ember.
Sok várakozás

Kézbesítve jelölés

Kézbesített Rendelések

Rendelés #91	Rendelés #92	Rendelés #103
Város: Subotica	Város: Subotica	Város: Subotica
Ulica: Tordanska 12	Ulica: Tordanska 12	Ulica: Tordanska 12
Összesen: 500 RSD	Összesen: 985 RSD	Összesen: 250 RSD
Kézbesítve	Kézbesítve	Kézbesítve
Kézbesítési idő: 18 perc 4 másodperc	Kézbesítési idő: 20 perc 35 másodperc	Kézbesítési idő: 2 perc
Megjegyzés:	Megjegyzés:	Megjegyzés:
Megjegyzés Mentése	Megjegyzés Mentése	Megjegyzés Mentése

Admin bejelentkezése "Kétlépcsős azonosítóval" :

Login

Email:

basicdavid79@gmail.com

Password:

Login

Kétlépcsős azonosítás

Kód:

619154

Ellenőrzés

Miután az admin belépett, hozzáadhat új adminisztrátort, valamint szerkesztheti a munkások és a felhasználók adatait.

Admin Felhasználó Munkás Étel Rendelés Üzenet Logout

Új kiszállító hozzáadása

Keresés kiszállító e-mail alapján

Kiszállító E-mail

john.doe@gmail.com

Keresztnév

John

Vezetéknév

Doe

E-mail

john.doe@gmail.com

Mentés Törölés

Az étel menüpont alatt hozzá adhat új ételt , szerkezhetheti a meglévőket , akciót is hozzá adhat , és törölheti a meglévő akciókat.

Új Étél Hozzáadása

Étel Kategória

Étel Neve

Étel Ára

Étel Mennyisége

Étel Leírása

Étel Hozzáadása

Akció Hozzáadása

Válasszon ételt

Kérem válasszon

Étel Kategória

Étel Neve

Étel Ára

Étel Mennyisége

Étel Leírása

Alías

Változtatások Mentése

Akció Hozzáadása

Válasszon akciót

Kérem válasszon

Akció Hozzáadása

Akció Törlése

A rendelések menü pont alatt láthatja a rendelések adatait és hozzá rendelhet egy futárt egy rendeléshez, és láthatja az összes rendelést.

Rendelések													
Rendelés ID	Felhasználó ID	Rendelés Dátuma	Város	Utca	Állapot	Ár	Megjegyzés	Telefonszám	Szállító ID	Admin ID	Szállítás Dátuma	Szállító Személy	Művelet
1	3	2024-06-18 20:30:23	Palic	vranjska	Kiszállítva	500		0600142949	4	4	2024-06-18 20:30:30	jól ▾	Hozzárendel
2	3	2024-06-18 00:00:00	Palic	vranjska	Kiszállítva	1485		0600142949	1		2024-06-18 17:57:26	jól ▾	Hozzárendel
3	3	2024-06-18 20:22:06	Palic	vranjska	Kiszállítva	500		0600142949	1	4	2024-06-18 20:22:11	jól ▾	Hozzárendel
4	3	2024-06-18 20:22:22	Palic	vranjska	Kiszállítva	1485		0600142949	1		2024-06-18 20:22:23	jól ▾	Hozzárendel
5	3	2024-06-18 20:30:35	Palic	vranjska	Kiszállítva	1485		0600142949	1		2024-06-18 20:30:38	jól ▾	Hozzárendel
6	3	2024-06-18 21:20:27	Palic	vranjska	Kiszállítva	3500		0600142949	1		2024-06-18 21:20:29	jól ▾	Hozzárendel
7	3	2024-06-18 20:35:23	Palic	vranjska	Kiszállítva	3500		0600142949	1		2024-06-18 20:35:24	jól ▾	Hozzárendel
8	3	2024-06-19 08:59:26	Palic	vranjska	Kiszállítva	3500		0600142949	1		2024-06-19 08:59:27	jól ▾	Hozzárendel
9	3	2024-06-19 08:36:18	Palic	vranjska	Kiszállítva	3500		0600142949	1		2024-06-19 08:59:15	jól ▾	Hozzárendel
10	3	2024-06-19 12:48:36	Palic	vranjska	Kiszállítva	3500		0600142949	1	4	2024-06-19 12:47:42	jól ▾	Hozzárendel

Az üzenet menü pont alatt láthatja a megrendeléshez és az ételhez hozzáfűzött értékelést és hozzászólásokat.

Delivery Ratings and Comments			
Delivery ID	Rating	Comment	Order ID
1	5	Szuper	53
1	5	Szuper2	53
1	5	Szuper2	53
1	5	Nagyon kedves a kiszállító.	55
1	5	Kedves volt.	66
1	5	Nagyon gyors kiszállítás.	98
1	5	Minden ok.	88
1	5	Kedves.	89
1	3	Lassuka volt	104
1	2	ngelne	106

Food Ratings and Comments		
Rating	Comment	Order ID
3	i loved it	67
5	Nagyon szupika	67
4	Szupi	67
5	Tetszik	67
3	Nagyon finom	68
5	I loved it.	69
5	Nagyon finom és friss étel volt. többször is rendelni fogok.	66
		60

5. Felhasznált irodalom

A felhasznált irodalomban a szerzők nevét ábécé sorrendben kell feltüntetni. Példa:

1. Brent R. Matzelle - PHPMailer, email küldés PHP-val
2. Dr. Zlatko Čović - upload-regiszter példa oldalak

W1. <https://github.com/PHPMailer/PHPMailer>

W2. <https://people.vts.su.ac.rs/~chole/wp/primeri-peldak/>