

Rendszerterv

1. Bevezetés:

A rendszerterv célja, hogy bemutassa a tervezett online ételrendelő platform fő vonásait, kiemelve annak alapvető céljaival és előnyeivel, mind az ügyfelek, mind az éttermek szempontjából. A rendszer kialakításának fő szándéka egy hatékony, egyszerű és átlátható ételrendelési élmény biztosítása mindkét érintett fél számára.

A rendszer célja:

A tervezett rendszer központi célja egy modern és felhasználóbarát online ételrendelő platform létrehozása. Az elgondolás mögötti fő motívum az, hogy lehetővé tegye az ügyfelek számára a könnyű és gyors ételrendelést, miközben az étterem számára egy hatékony rendszer álljon rendelkezésére a rendelések kezelésére. A rendszer által kínált funkciók terjednek a regisztrációtól a kiszállításig, azaz az ételrendelési folyamat teljes spektrumát lefedik.

2. A rendszer előnyei:

A rendszer tervezett előnyei közül kiemelhetjük:

1. Gyors és hatékony rendelési folyamat:
Az ügyfeleknek lehetőségük van gyorsan és hatékonyan leadni rendeléseiket, minimalizálva a felhasználói erőfeszítéseket és időráfordítást.
2. Rugalmas étlap kezelés:
Az étterem számára biztosítjuk a kínálatuk könnyű és rugalmas kezelését, beleértve az ételek frissítését és újak hozzáadását.
3. Rendelések nyomonkövetése:
Az ügyfelek és étterem egyszerűen nyomon követhetik a rendeléseket, elősegítve az átláthatóságot és a folyamatos kommunikációt.
4. Felhasználói visszajelzések és értékelések:
A rendszer lehetőséget kínál a felhasználók számára értékeléseik és visszajelzéseik megosztására, ezáltal elősegítve a minőségellenőrzést és a folyamatos fejlesztést.
5. Biztonság és Adatvédelem:
Az ügyfelek személyes adatainak hatékony védelme érdekében erős biztonsági intézkedéseket alkalmazunk, és szigorúan betartjuk az adatvédelmi előírásokat.

Ezen előnyök összességében hozzájárulnak a rendszer célkitűzéseinek eléréséhez, mégpedig egy hatékonyabb és kényelmesebb ételrendelési folyamat révén mind az ügyfelek, mind az étterem számára.

3. Rendszer Architektúra:

2.1 Kliens-szerver architektúra:

Az ételrendelő rendszerünk a hagyományos kliens-szerver architektúra alapján működik. A kliens a webböngésző révén kommunikál a PHP-alapú szerverrel, amely a kérésekre válaszokat küld vissza.

2.2 Technológiai Stack:

- Frontend:
 - A felhasználói felület fejlesztése HTML, CSS, JavaScript nyelven történt.
- Backend:
 - Szerveroldali logika implementálása PHP nyelven.
- Adatbázis:
 - Az ételrendelő rendszer adatbázisát a PHPMyAdmin felületén alakítottuk ki. A MySQL adatbázist használjuk az ételrendelések, felhasználói profilok és éttermek adatainak tárolására. Az adatbázis struktúrája külön pontban kerül bemutatásra.

2.3 Authentikáció és Biztonság:

- A biztonság javítása érdekében a kommunikáció titkosítása HTTPS protokollal.
- PHP nyelv belső biztonsági funkcióinak felhasználása.

2.4 Mikroszolgáltatások (opcionális):

- A rendszer egyszerűségének és méreteinek megfelelően, a funkciókat egy egységes PHP alkalmazásban kezeljük, nincs külön mikroszolgáltatások használata.

2.5 Adatbázis és Kapcsolatok:

- Az ételrendelő rendszerünk az adatbázis kezeléséhez a PHPMyAdmin szolgáltatását használja, amely MySQL-alapú adatbázist biztosít az ételrendelések, felhasználói profilok és éttermek részletes tárolásához. Az adatbázis struktúrája külön pontban kerül bemutatásra.

2.6 Felhasználói Interfész:

- A felhasználói felület PHP nyelv segítségével lett kialakítva kialakított rezponzív design-nal, hogy a rendszer kényelmesen használható legyen a böngészőkön és mobil eszközökön egyaránt.

2.7 Követelmények:

- A rendszer minimális szoftver- és hardverkövetelményeinek megfelelően kialakítva.

2.8 Teljesítmény és Skálázhatóság:

- A rendszer egyszerűsége miatt a teljesítmény és skálázhatóság nem kritikus tényezők.

2.9 Biztonság és Adatvédelem:

- A rendszer biztonságáért a PHPMyAdmin felületén egyszerű, de hatékony lépéseket alkalmazunk:
- Form Adatok Validálása:
A PHPMyAdmin segítségével ellenőrizzük és megerősítjük, hogy a felhasználók által beküldött adatok megfelelőek és biztonságosak legyenek. Így csak a helyes adatokat engedjük át.
- **Biztonságos Lekérdezések:**
Az adatbázisban történő lekérdezéseknél odafigyelünk arra, hogy biztonságos módszereket alkalmazzunk. Például, hogy megelőzzük az SQL injection támadásokat, paraméterekkel vagy előre elkészített utasításokkal védekezünk.

2.10 Rendszer Vezérlők és Szolgáltatások:

- A PHP alkalmazás vezérlői és szolgáltatásai, például a rendelések kezelése, étlap frissítése és felhasználói adatok kezelése.

Az ételrendelő rendszer architektúrája PHP nyelvű szerveroldali fejlesztésre épül, az egyszerűség és funkcionalitás szem előtt tartásával.

4. Adatbázis tervezés

- Táblák, mezők és kapcsolatok leírása.
- Adatbázis diagram.

5. Felhasználói funkciók

Az alkalmazás szempontjából kulcsfontosságú használati eseteket a következők szerint részletezhetjük:

Regisztráció és Bejelentkezés:

- *Felhasználó regisztrációja:*

A felhasználó kitölti a regisztrációs űrlapot, megadva szükséges személyes információikat (név, e-mail, jelszó stb.).

Az alkalmazás ellenőrzi a megadott adatok helyességét és egyedülállóságát.

Sikeres regisztráció esetén a felhasználó kap egy megerősítő e-mailt vagy egy visszaigazoló üzenetet.

- *Bejelentkezés:*

A regisztrált felhasználó bejelentkezik az alkalmazásba, megadva az egyedi belépési adatait.

Az alkalmazás ellenőrzi az azonosítót és a jelszót.

Sikeres bejelentkezés után a felhasználót átirányítjuk a főoldalra.

Étel kiválasztása és Rendelése:

- *Étlap böngészése:*

A felhasználó áttekinti az étterem étlapját, ahol láthatják a különböző ételeket és azok árait.

A kategóriák és szűrők segítségével könnyen navigálnak az étlapon.

- *Kosárba helyezés:*

A felhasználó kiválasztja az ételeket, amiket rendelni szeretne, majd hozzáadja azokat a kosarához.

A kosárban lehetőség van a mennyiség módosítására, valamint az ételek eltávolítására.

- *Rendelés leadása:*

A felhasználó elküldi a kosár tartalmát az étterem felé.

Az alkalmazás automatikusan visszaigazolja a rendelést és értesítést küld a sikeres feldolgozásról.

Rendelés Követése és Értékelése:

Rendelés Követése:

A felhasználó nyomon követheti a rendelése állapotát a weboldalon.

Az alkalmazás automatikusan értesítést küld a kiszállítás előrehaladásáról.

Értékelés és Visszajelzés:

A felhasználó az étel értékelése mellett hozzászólhatnak is a rendelt ételekhez.

6. Admin funkciók:

- *Étlap kezelése:*

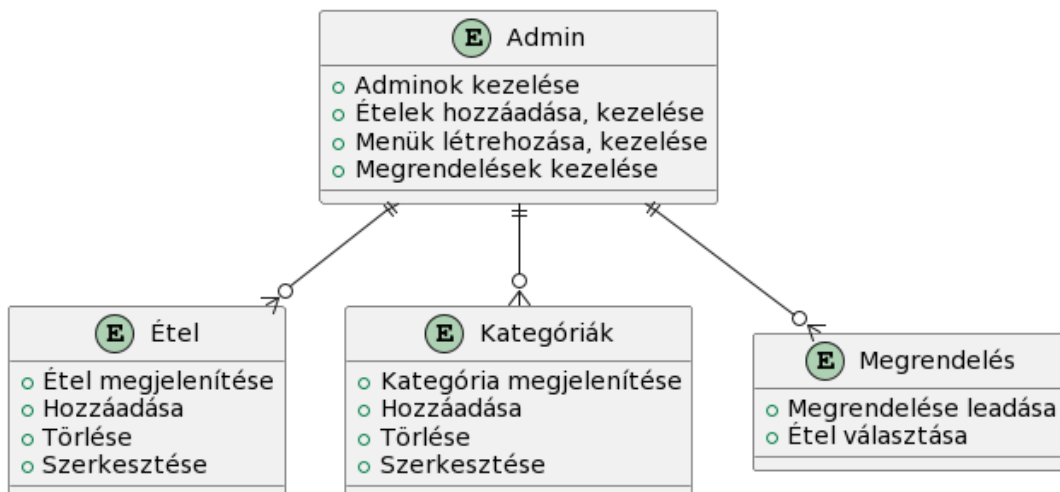
Az étterem adminja frissíti és kezeli az étlapot, hozzáadva, módosítva vagy törölve ételeket.

- *Rendelések kezelése:*

- Az admin nyomon követi és frissíti a beérkező rendeléseket, valamint kezeli azok státuszát.

- *Felhasználói adatok kezelése:*

- Az admin kezeli a felhasználói adatokat és profilbeállításokat, beleértve a regisztrált felhasználók adatainak módosítását vagy törlését.



7. Használati esetek:

Felhasználó:

- Regisztráció és Bejelentkezés:

A felhasználó a regisztráció során személyes adatait, mint például nevét, e-mail címét és jelszavát, megadva hoz létre egy fiókot.

A regisztrációt követően a felhasználó bejelentkezik a rendszerbe az egyedi belépési adataival.

- Étel kiválasztása és Kosárba helyezése:

Az felhasználó böngészi az étterem kínálatát, választ az ételek közül, majd a kiválasztott ételeket hozzáadja a virtuális kosarához.

A kosárban lehetőség van módosítani a mennyiséget és eltávolítani az ételleket.

- Megrendelés:

Az ügyfél a kosárában lévő ételek után továbblép a megrendelés folyamatához.

A megrendelés során megadja a szállítási címet és véglegesíti a rendelést.

- Rendelés követése és Visszajelzés írása:

Az ügyfelek az alkalmazásban nyomon követik rendelésüket, és értesülnek a rendelésük aktuális állapotáról.

A rendelés teljesítése után az ügyfelek értékelést írnak az ételekről és a szolgáltatásról.

Admin:

- Regisztráció és Profil kezelése:

Az admin regisztrál a rendszerbe, és kezeli saját profilját, beleértve a felhasználói adatokat és jelszót.

- Étlap frissítése és Kezelése:

Az admin frissíti az étlapot, hozzáadva, módosítva vagy törölve ételeket az étterem kínálatából.

- Rendelések fogadása és Státusz frissítése:

Az admin fogadja és kezeli a beérkező rendeléseket, frissíti a rendelési státuszt, és értesíti az ügyfeleket a változásokról.

8. Felhasználói felület:

Főoldal:

- Az étterem logójával és nevével üdvözljük a látogatókat.
- Kiemelt ajánlatok és speciális ételkategóriák mutatása.
- Gyors hozzáférés a teljes étlaphoz és a legújabb ételekhez.

Étlap:

- Kategóriákba rendezett ételek listája, mindegyikhez tartozó leírással és képpel.
- Szűrési és keresési lehetőségek a könnyebb böngészés érdekében.

Rendelési oldal:

- A kiválasztott ételek megjelenítése egy virtuális kosárban.
- Lehetőség a kosár tartalmának módosítására (mennyiség, eltávolítás).
- Szállítási információk megadása, mint például a cím és a kívánt időpont.

Profil oldal:

- A felhasználói fiók kezelése, beleértve a személyes adatok és jelszó módosítását.
- Rendelési előzmények megtekintése és értékelések írása.

Admin Dashboard:

- Rendelések kezelése:

Új rendelések megtekintése és státuszuk frissítése.

- Étlap kezelése:

Ételek hozzáadása, módosítása vagy törlése.

- Felhasználói adatok kezelése:

Regisztrált felhasználók adatainak megtekintése és kezelése.

- Statisztikák és riportok:

Tevékenységi statisztikák és riportok megtekintése az adminisztrátor számára.

9. Biztonság és Adatvédelem:

- SSL használata.

- Felhasználói jelszavak biztonságos tárolása.
- GDPR-megfelelőség.

10. Tesztelési Stratégia:

Egységtesztek:

- Cél: Az egyes komponensek (függvények, metódusok) helyes működésének ellenőrzése.
- Mit tesztel: Függvényeket, osztályokat, metódusokat.

Funkcionális Tesztek:

- Cél: Az alkalmazás funkcionálisának tesztelése.
- Mit tesztel: Felhasználói scénáriókat, funkciókat (pl., regisztráció, bejelentkezés, rendelés leadása).

Teljesítménytesztek:

- Cél: Az alkalmazás teljesítményének ellenőrzése különböző terhelések mellett.
- Mit tesztel: Az alkalmazás válaszidejét, terhelés alatti stabilitását.

Részletes leírás:

Egységtesztek:

- Felhasználó regisztráció funkció:
 - Teszt, hogy a regisztrációs űrlap helyesen validálja-e a felhasználói adatokat.
 - Ellenőrizés, hogy a regisztrációs funkció helyesen hozza létre az új felhasználót az adatbázisban.

Funkcionális tesztek:

- Rendelés leadása:
 - Rendelési folyamat tesztelése a kiválasztott ételek kosárba helyezésétől a visszaigazolásig.
 - Rendelés státusz ellenőrzése az adminisztrátori felületen.

Teljesítménytesztek:

- Regisztráció Terheléses Teszt:
 - Hozz létre párhuzamosan futtatott regisztrációs kéréseket, és nézd meg, hogyan reagál az alkalmazás.
 - Mérjük meg a válaszidőt és az esetleges hibákat.

Tesztelési szekvencia:

Egységtesztek futtatása:

- Cél: Bizonyítani, hogy az egyes részek helyesen működnek.
- Szekvencia:
 - Kódolás és Egységtesztelés: Minden új kódelemhez egységteszteket írunk.
 - Automatikus Egységteszt Futtatása: A CI/CD rendszer automatikusan futtatja az egységteszteket minden kódmódosítás után.
 - Visszajelzés: Hibák vagy meghiúsulások esetén azonnal kapunk visszajelzést.

Funkcionális Tesztek Futtatása:

- Cél: Ellenőrizni az alkalmazás üzleti scénárióinak megfelelőségét.
- Szekvencia:
 - Funkcionális Tesztesetek Kialakítása: Fontos üzleti scénáriókra tervezett tesztesetek készítése.
 - Tesztek Végrehajtása: A tesztesetek futtatása az alkalmazás fejlesztési szakaszának különböző pontjain.
 - Visszajelzés: Hibajelentések és visszajelzések begyűjtése a tesztek eredményeiről.

Teljesítménytesztek Futtatása:

- Cél: Ellenőrizni az alkalmazás válaszidejét és stabilitását terhelés alatt.
- Szekvencia:
 - Teljesítmény Tesztterv Kidolgozása: Terhelési forgatókönyvek és tesztek tervezése.
 - Tesztek Futtatása: Terhelés alatti teljesítménytesztek végrehajtása.
 - Eredmények Elemzése: Válaszidők, hibák, erőforrások felhasználása elemzése.
 - Optimalizáció és Hibajavítás: Az eredmények alapján az alkalmazás optimalizálása és esetleges hibák kijavítása.

Hibajavítás és Optimalizáció:

- Cél: Javítani és optimalizálni az alkalmazást a teszteredmények alapján.
- Szekvencia:
 - Hibajavítás: Az egységtesztek és a funkcionális tesztek hibáinak javítása.
 - Optimalizáció: Az alkalmazás teljesítményének és skálázhatóságának javítása.
 - Tesztek Újrafuttatása: A módosítások után újra lefuttatjuk az egység- és funkcionális teszteket.

Ismételt Tesztek:

- Cél: Biztosítani, hogy a módosítások ne vezessenek be új hibákat.
- Szekvencia:
 Egység- és Funkcionális Tesztek Ismételt Futtatása: Ellenőrizzük, hogy az új kódmódosítások nem rontottak el korábbi funkciókat.
 Visszajelzés: Ha új hibák jelennek meg, visszajelzést kapunk és a folyamatot újból elindítjuk.

11. Bővítési Lehetőségek:

Lehetőségek a rendszer bővítésére (pl. előfizetési rendszer, értékelési rendszer).

12. Technológiai Stack:

Frontend Technológiai Stack:

- Kódolási Környezet: Visual Studio Code.
- JavaScript Keretrendszer: **Nincs említve specifikusan a frontend részen, de ha PHP a backend, akkor az AJAX segítségével lehet aszinkron kommunikációt végezni a szerverrel.**
- Verziókezelés: GitHub - a csapat közös munkájának koordinálására és kód verziókezelésére.
- Fejlesztési nyelv: **JavaScript, és a frontend technológiáktól függően (React, Angular, stb.).**

Backend Technológiai Stack:

- Szerveroldali Környezet: Apache szerver.
- Adatbázis Kezelő: PHPMyAdmin - MySQL adatbázis.
- Backend Nyelv: PHP - az üzleti logika és a szerveroldali műveletek megvalósítására.
- Autentikáció és Engedélyezés: **JWT (JSON Web Token) és Passport.js - a felhasználói autentikáció és jogosultságok kezelésére.**
- API Dokumentáció: **Nem említve specifikusan, de Swagger vagy OpenAPI lehet hasznos a szerveroldali API dokumentálásához.**
- Tesztelés: **Mocha vagy Jest - egységtesztek és integrációs tesztek.**

Verziókezelés:

- GitHub: A csapat együttműködésének és kódverziókezelésének a helye.

13. Dokumentáció:

API dokumentáció.

Adatbázis séma és kapcsolati diagramok.

```
Table tbl_megrendeles {  
  
  id INT [primary key]  
  
  etel_id INT  
  
  etel VARCHAR(150)  
  
  ar DECIMAL(10,2)  
  
  mennyiseg INT  
  
  teljes DECIMAL(10,2)  
  
  rendeles_datum DATETIME  
  
  statusz VARCHAR(50)  
  
  rendelo_nev VARCHAR(150)  
  
  rendelo_elerhetoseg VARCHAR(20)  
  
  rendelo_email VARCHAR(150)  
  
  rendelo_cim VARCHAR(255)  
  
}
```

```
Table tbl_etel {  
  
    id INT [primary key]  
  
    nev VARCHAR(100)  
  
    leiras TEXT  
  
    ar DECIMAL(10,2)  
  
    kep_nev VARCHAR(255)  
  
    kategoria_id INT  
  
    jelleg VARCHAR(10)  
  
    active VARCHAR(10)  
  
}
```

```
Table tbl_kategoria {  
  
    id INT [primary key]  
  
    nev VARCHAR(100)  
  
    kep_neve VARCHAR(255)  
  
    jelleg VARCHAR(10)  
  
    van_e VARCHAR(10)  
  
}
```

```
Table tbl_admin {  
  
    admin_id INT [primary key]
```

```
teljes_nev VARCHAR(100)

felhasznev VARCHAR(100)

jelszo VARCHAR(255)

}

Table tbl_statusz {

    id INT [primary key]

    statusz VARCHAR(50)

}

Ref: tbl_etel.id < tbl_kategoria.id
```