[**https://www.dynamsoft.com/codepool/nfc-barcode-scanner.html**](https://www.dynamsoft.com/codepool/nfc-barcode-scanner.html)

**Felhasználó (Users)**

* id (PK)
* first\_name
* last\_name
* email
* password\_hash
* user\_type (tanár/diák/admin)
* class\_id (tanároknél NULL)
* created\_at
* discount?????????

**Étel (Meals)**

* id (PK)
* név
* leírás (kalória)
* allergén\_info
* is\_available
* created\_at

**Rendelés (Orders)**

* id (PK)
* user\_id (FK → Users)
* meal\_id (FK → Meals)
* order\_date (melyik napra szól a rendelés)
* status (rendelt / lemondva / kifizetve)
* cancelled
* cancelled\_at
* created\_at

**Értékelés (Ratings)**

* id (PK)
* user\_id (FK → Users)
* meal\_id (FK → Meals)
* rateing (1–5)
* comment
* created\_at

**Class**

* id (PK)
* name
* created\_at

**Payment**

* id
* user\_id
* order\_date
* amount
* status
* payment\_date
* created\_at

**Price**

* student\_price
* teacher\_price
* valid\_from
* valid\_until
* created\_at

-- Osztályok tábla

CREATE TABLE classes (

id SERIAL PRIMARY KEY,

name VARCHAR(50) NOT NULL UNIQUE,

grade\_level INTEGER,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

-- Felhasználói típus enum

CREATE TYPE user\_type\_enum AS ENUM ('student', 'teacher', 'admin');

-- Felhasználók tábla

CREATE TABLE users (

id SERIAL PRIMARY KEY,

first\_name VARCHAR(50) NOT NULL,

last\_name VARCHAR(50) NOT NULL,

email VARCHAR(100) UNIQUE NOT NULL,

password\_hash VARCHAR(255) NOT NULL, -- Jelszó hash-elés biztonsági okokból

user\_type user\_type\_enum NOT NULL DEFAULT 'student',

class\_id INTEGER REFERENCES classes(id) ON DELETE SET NULL,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

CONSTRAINT teacher\_no\_class CHECK (

(user\_type = 'teacher' AND class\_id IS NULL) OR

(user\_type != 'teacher')

)

);

-- Ételek tábla

CREATE TABLE meals (

id SERIAL PRIMARY KEY,

name VARCHAR(100) NOT NULL,

description TEXT,

calories INTEGER, -- Kalória egész számként

teacher\_price DECIMAL(10, 2) NOT NULL,

student\_price DECIMAL(10, 2) NOT NULL,

allergen\_info TEXT,

is\_available BOOLEAN DEFAULT TRUE,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

-- Rendelések tábla

CREATE TABLE orders (

id SERIAL PRIMARY KEY,

user\_id INTEGER NOT NULL REFERENCES users(id) ON DELETE CASCADE,

meal\_id INTEGER NOT NULL REFERENCES meals(id) ON DELETE CASCADE,

order\_date DATE NOT NULL,

status VARCHAR(20) DEFAULT 'ordered', -- ordered, cancelled, paid

cancelled BOOLEAN DEFAULT FALSE,

cancelled\_at TIMESTAMP,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UNIQUE(user\_id, order\_date) -- Egy felhasználó csak egy ételt rendelhet naponta

);

-- Értékelések tábla

CREATE TABLE ratings (

id SERIAL PRIMARY KEY,

user\_id INTEGER NOT NULL REFERENCES users(id) ON DELETE CASCADE,

meal\_id INTEGER NOT NULL REFERENCES meals(id) ON DELETE CASCADE,

rating INTEGER CHECK (rating >= 1 AND rating <= 5) NOT NULL,

comment TEXT,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UNIQUE(user\_id, meal\_id) -- Egy felhasználó csak egyszer értékelhet egy ételt

);

-- Fizetések tábla

CREATE TABLE payments (

id SERIAL PRIMARY KEY,

user\_id INTEGER NOT NULL REFERENCES users(id) ON DELETE CASCADE,

order\_date DATE NOT NULL,

amount DECIMAL(10, 2) NOT NULL,

status VARCHAR(20) DEFAULT 'pending', -- pending, completed, failed

payment\_date TIMESTAMP,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UNIQUE(user\_id, order\_date) -- Egy felhasználónak naponta csak egy fizetése lehet

);

-- Árbeállítások tábla (ha szeretnéd historikusan követni az árváltozásokat)

CREATE TABLE price\_settings (

id SERIAL PRIMARY KEY,

student\_price DECIMAL(10, 2) NOT NULL,

teacher\_price DECIMAL(10, 2) NOT NULL,

valid\_from DATE NOT NULL DEFAULT CURRENT\_DATE,

valid\_until DATE,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

**Dokumentáció szerkezete**

**1. Címoldal**

* Projekt címe (pl. „Iskolai étkezéskezelő rendszer”)
* Készítő neve
* Dátum
* Vizsgafeladat megnevezése

**2. Bevezetés**

* Röviden a projekt célja: miért készült, milyen problémát old meg
* Példa: *„**Az iskolai étkezések kezelése gyakran papíron történik, ami nehézkes. A rendszer célja egy átlátható, könnyen kezelhető felület biztosítása a menü megtekintésére, rendelésre, lemondásra és fizetésre.”*

**3. Követelmények**

* **Funkcionális követelmények** (pl. menü megtekintése, rendelés, lemondás, értékelés)
* **Nem funkcionális követelmények** (pl. reszponzív felület, adatbiztonság, egyszerű kezelhetőség)

**4. Rendszerterv**

* **Architektúra**: frontend + backend + adatbázis rövid bemutatása
* **Felhasználói szerepkörök**: diák, szülő, admin (ki mit lát, mit tehet)
* **Adatbázis-terv (ER-diagram)**

**5. Megvalósítás**

* Használt technológiák (pl. HTML, CSS, JavaScript, PHP/React, SQLite/MySQL)
* Felhasználói felület rövid bemutatása (képernyőképekkel, ha lesz kész)
* Példák a működésre (pl. rendelés folyamata, lemondás folyamata)

**6. Tesztelés**

* Hogyan tesztelted a programot?
* Példa:
  + Bejelentkezés helyes/jelszóhibás felhasználóval
  + Ebéd rendelése és lemondása
  + Értékelés rögzítése
* Esetleg egy táblázat, hogy „Bemenet → Elvárt kimenet → Kapott kimenet”

**7. Összefoglalás**

* Rövid értékelés: mit sikerült megvalósítani, mi lenne még fejleszthető
* Példa: *„A rendszer teljesíti az alapfunkciókat, a jövőben kiegészíthető mobilalkalmazással és QR-kód alapú ételkiadással.”*

**8. Mellékletek (opcionális)**

* ER-diagram
* Képernyőképek a kész programról
* Kódrészletek