Uživatelská a administrátorská příručka aplikace “ReserveIt“

Contents

[Úvod 2](#_Toc198830721)

[Uživatelská příručka 2](#_Toc198830722)

[1. Rezervace stolu 2](#_Toc198830723)

[2. Úprava nebo zrušení rezervace 2](#_Toc198830724)

[3. Přihlášení do administrace 3](#_Toc198830725)

[4. Správa rezervací 3](#_Toc198830726)

[5. Správa stolů 3](#_Toc198830727)

[6. Správa menu 3](#_Toc198830728)

[6. Kalendář rezervací 4](#_Toc198830729)

[7. Statistiky 4](#_Toc198830730)

[Administrátorská příručka 4](#_Toc198830731)

[1. Technické požadavky 4](#_Toc198830732)

[2. Instalace a spuštění 5](#_Toc198830733)

[**3. Struktura projektu** 6](#_Toc198830734)

[4. Databázová správa 6](#_Toc198830735)

[5. Autentizace a oprávnění 6](#_Toc198830736)

[6. Logování a ladění 6](#_Toc198830737)

[7. Nasazení 7](#_Toc198830738)

[8. Zálohování a obnova dat 7](#_Toc198830739)

[9. Bezpečnostní opatření 7](#_Toc198830740)

# Úvod

Tato příručka slouží jako přehledný průvodce pro koncové uživatele i administrátory systému online rezervací restaurací. Systém je navržen tak, aby umožnil snadnou správu rezervací, stolů, jídelních lístků a statistik bez potřeby složitých procesů.

Uživatelé mohou pohodlně provádět rezervace online a správci restaurace mají k dispozici administrativní rozhraní pro kompletní kontrolu nad provozem.

# Uživatelská příručka

## 1. Rezervace stolu

1. Otevřete veřejnou rezervační stránku restaurace.
2. Formulář je rozdělený do tří kroků které vás povedou
3. V první části vybereme datum, pocet osob a čas rezervace
4. V dalších dvou krocích vyplníme formulář:
   * Jméno
   * Příjmení
   * E-mail
   * telefon
   * Poznámka (nepovinné)
5. Systém automaticky nabídne dostupné časy podle obsazenosti.
6. Po odeslání obdržíte e-mail s potvrzením a odkazem pro úpravu/zrušení rezervace.

## 2. Úprava nebo zrušení rezervace

1. Otevřete odkaz z e-mailu.
2. Můžete:
   * Změnit datum, čas nebo počet osob
   * Zrušit rezervaci

### 3. Přihlášení do administrace

1. Otevřete administrační URL adresu (např. /admin).
2. Zadejte přihlašovací e-mail a heslo.
3. Po přihlášení uvidíte dashboard s grafy a statistikami, notifikacemi a rezervacemi.

### 4. Správa rezervací

* **Zobrazení rezervací:** Filtrujte podle data, času, stavu.
* **Stavy rezervací:**
  + pending – čeká na potvrzení
  + confirmed – potvrzená rezervace
* **Úpravy rezervace:** Změňte čas, počet osob, poznámku nebo stav.
* **Mazání:** Rezervaci lze trvale odstranit.

### 5. Správa stolů

* Otevřete sekci „Stoly“
* Přidejte nebo upravte stůl:
  + Číslo stolu
  + Kapacita (počet osob)
  + Lokace (např. terasa, interiér)
* Stoly lze přetahovat v rozhraní pro vizuální správu

### 6. Správa menu

* Menu je rozděleno do záložek (např. Denní menu, Nápoje)
* Každý tab muze obsahovat menu které lze označit, jestli je aktivní či ne
* Aktivní menu se budou zobrazovat přímo na stránkách restaurace ¨
* Položky v menu se dají pohodlně přesouvat pomocí přetažení
* Pozice položky definuje i to jak se bude zobrazovat na stránkách restaurace
* Do menu přidávejte položky:
  + Název
  + Popis
  + Cena
  + Alergeny

### 6. Kalendář rezervací

* Najdete v sekci „Kalendář“
* Zobrazí přehled stolů a jejich rezervací po hodinách
* Možnost přepínat dny a kontrolovat dostupnost

### 7. Statistiky

* Grafy zobrazují:
  + Počet rezervací v čase
  + Vytížení stolů

# Administrátorská příručka

Tato příručka poskytuje technické informace a postupy pro správu a provoz rezervačního systému restaurací. Je určena pro administrátory a vývojáře, kteří se starají o nasazení, údržbu a rozvoj systému.

## 1. Technické požadavky

#### 1.1. Frontend

* **Technologie:** React, Tailwind CSS
* **Požadavky:**
  + Node.js (verze 14 nebo vyšší)
  + npm (verze 6 nebo vyšší)

#### 1.2. Backend

* **Technologie:** NestJS
* **Požadavky:**
  + Node.js (verze 14 nebo vyšší)
  + npm (verze 6 nebo vyšší)

#### 1.3. Databáze

* **Služba:** Supabase (PostgreSQL)

#### 2.4. Ostatní nástroje

* Git (pro správu verzí)
* Webový prohlížeč (Chrome, Firefox, Edge)

## 2. Instalace a spuštění

#### 2.1. Klonování repozitáře

bash

git clone https://github.com/stepan-kobrle/rezervacni-system.git

cd rezervacni-system

#### 2.2. Nastavení prostředí

Vytvořte soubor .env v kořenovém adresáři a nastavte potřebné proměnné prostředí pro frontend a backend, včetně přístupových údajů k Supabase.

#### 2.3. Instalace závislostí

**Frontend:**

* bash
* cd frontend
* npm install

**Backend:**

* bash
* cd backend
* npm install

#### 3.4. Spuštění aplikace

**Frontend:**

* bash
* npm run dev

Aplikace poběží na http://localhost:3000.

**Backend:**

* bash
* npm run start:dev

API poběží na <http://localhost:4000>.

## **3. Struktura projektu**

* /frontend – React aplikace s Tailwind CSS
* /backend – NestJS aplikace
* backend/prisma – schéma databáze a migrace
* /docs – dokumentace projektu

## 4. Databázová správa

#### 4.1. Supabase

* Přístup k databázi přes Supabase dashboard
* Správa tabulek, pohledů a funkcí
* Nastavení autentizace a pravidel přístupu

#### 4.2. Prisma

* Definice schématu v prisma/schema.prisma
* Migrace databáze pomocí příkazů:
  + npx prisma migrate dev
* Generování Prisma Clienta:
  + bash
  + npx prisma generate

## 5. Autentizace a oprávnění

* **Supabase Auth:** Správa uživatelů a autentizace
* **Role:**
  + ADMIN – plný přístup ke všem funkcím
  + MANAGER – správa menu, stolů a rezervací
  + EMPLOYEE – přístup k rezervacím a kalendáři
* **Ověřování přístupů:** Implementováno pomocí Guardů a Middleware v NestJS

## 6. Logování a ladění

* **Backend:**
  + Použití NestJS loggeru pro logování událostí
  + Middleware pro zachytávání a zpracování chyb
* **Frontend:**
  + Zobrazení chybových hlášek ve formulářích
  + Notifikace pro uživatele při chybách

## 7. Nasazení

* **Frontend:** Možnost nasazení na Vercel nebo Netlify
* **Backend:** Nasazení na platformy podporující Node.js aplikace (např. Heroku, Render)
* **Databáze:** Supabase jako cloudové řešení PostgreSQL

## 8. Zálohování a obnova dat

* **Supabase:** Pravidelné zálohování databáze přes Supabase dashboard
* **Obnova dat:** Možnost obnovení databáze ze záloh v případě potřeby

## 9. Bezpečnostní opatření

* **Šifrování:** Použití HTTPS pro zabezpečení komunikace
* **Ochrana dat:** Hashování hesel uživatelů pomocí bcrypt
* **Pravidla přístupu:** Nastavení RLS (Row Level Security) v Supabase pro omezení přístupu k datům