

## Úkol pro .NET Fullstack Developer (Razor Pages)

Tímto úkolem chceme zjistit, jak přemýšlíte o architektuře a jak píšete čitelný kód. Zároveň nás zajímá, jak řešíte zabezpečení a zda rozumíte tomu, co vytváříte.

Odhadovaná časová náročnost je 3 až 6 hodin.

### Zadání

Vytvořte jednoduché řešení složené ze dvou projektů:

1. **ASP.NET Core Web API** – zajišťuje autentizaci a data
2. **ASP.NET Core Razor Pages aplikace** – webové rozhraní pro uživatele

Razor Pages aplikace komunikuje s API výhradně přes HTTP.

### Web API

API bude obsahovat:

#### 1. Endpoint pro přihlášení

- Přijímá username a password
- Ověření uživatele může být hardcoded
- Při úspěšném přihlášení vrátí Bearer token (např. JWT)

#### 2. Zabezpečený endpoint

- Přístupný pouze s platným Bearer tokenem
- Vrací jednoduchá data (statická nebo generovaná)

#### Požadavky na API

- ASP.NET Core Web API
- Standardní zabezpečení pomocí Bearer tokenu
- Není nutná databáze, neřešte refresh tokeny, role ani pokročilé scénáře

## Razor Pages aplikace

### Přihlašovací stránka

- Formulář (username, password, button)
- Po odeslání se zavolá přihlašovací endpoint a získá se token
- Token uložte a používejte při volání zabezpečeného API endpointu

### Zabezpečená stránka

- Přístupná pouze přihlášenému uživateli
- Zavolá zabezpečený API endpoint a zobrazí vrácená data

## Technické požadavky

- .NET 8+ (C#)
- Razor Pages (ne React, Vue, apod.)
- Vlastní CSS (bez Tailwindu, Bootstrapu apod.)
- Použijte BEM pro názvy tříd, žádné inline styly

### Co není cílem

- Databáze
- Složitá autentizace (role, refresh tokeny)
- Docker, CI/CD

## README.md

Součástí odevzdání musí být stručné vysvětlení:

- Jak projekt spustit
- Testovací přihlašovací údaje

### Odevzdání

- Vytvořte vlastní repository
- Projekt musí jít spustit bez dalších úprav
- Pošlete nám odkaz na repo

Děkujeme, že jste si na úkol udělal/a čas.