# Dokumentace k projektu: Blogová stránka

## 1. Úvod

Tato dokumentace popisuje projekt blogové stránky, která umožňuje uživatelům prohlížet a spravovat blogy na základě jejich rolí (admin/editor). Projekt byl vytvořen za použití technologií React, Redux, Bootstrap a JavaScript. Data jsou načítána z JSON souborů a spravována pomocí Redux store.

## 2. Popis aplikace a jejích funkcí

### Hlavní funkce aplikace

Zobrazení blogů na hlavní stránce: Na hlavní stránce se zobrazují náhodně vybrané blogy. Uživatel může použít filtraci podle kategorie nebo vyhledávání blogů podle jejich titulku.

Detail blogu: Každý blog má možnost zobrazit detail, kde je dostupný podrobný obsah článku.

Přihlášení uživatele: Přihlašovací formulář umožňuje uživateli přihlásit se podle své role (admin nebo editor).

### Admin funkce

Přehled uživatelů a blogů: Admin má přístup ke kartám \*Users\* a \*Blogs\* v dashboardu.

Users: Seznam všech uživatelů s možností editace a mazání.

Blogs: Přehled všech blogů s možností editace a mazání.

### Editor funkce

Správa vlastních blogů: Editor má přístup pouze ke svým blogům, které může upravovat nebo mazat.

### Obecné funkce přihlášených uživatelů

Settings: Možnost upravit přihlašovací údaje prostřednictvím modálního okna.

Logout: Odhlášení uživatele.

Zpět na stránku: Přesměrování na hlavní stránku.

## 3. Technický popis řešení

### Architektura aplikace

Aplikace je postavena jako jednostránková (SPA) za použití React a Redux. Rozhraní je vytvořeno pomocí Bootstrapu, což zajišťuje responzivitu a moderní vzhled.

### Použité technologie

React: Pro tvorbu komponent a interaktivního uživatelského rozhraní.

Redux: Správa globálního stavu aplikace (uživatelé, blogy).

Bootstrap: Pro stylování a zajištění responzivity.

JavaScript: Logika aplikace.

JSON soubory (post.json a users.json): Zdrojová data pro blogy a uživatele.

### Struktura kódu

src/components: Obsahuje React componenty

src/data: obsahuje JSON soubory

src/pages: objsahuje jednotlivé stránky

src/redux:  
 - reducers: Definice funkcí pro správu stavu (uživatelé, blogy).  
 - actions: Akce pro Redux, např. načítání dat nebo editace.   
 - store.js: Centrální úložiště pro Redux.

### Datové toky

1. Načítání dat: Data o uživatelích a blozích se načítají z JSON souborů (`posts.json`, `users.json`) během inicializace aplikace.

2. Redux store: Ukládá stav aplikace, například přihlášeného uživatele, seznam blogů a uživatelů.

3. Komunikace komponent: Komponenty získávají data z Redux store a aktualizují je prostřednictvím dispatch akcí.

## 4. Závěr

Tento projekt poskytuje plně funkční blogovou platformu s rozdělením rolí uživatelů. Díky modularitě kódu a využití Reduxu je aplikace snadno rozšiřitelná o nové funkce, jako jsou pokročilá správa kategorií nebo integrace s databází.