Dokumentace k projektu: Online-store

Zabrodin Maksim

Obsah

[**Popis aplikace** 1](#_Toc188883717)

[Funkce aplikace 1](#_Toc188883718)

[Obrazovky aplikace 2](#_Toc188883719)

[**Technický popis** 8](#_Toc188883720)

[Použité technologie 8](#_Toc188883721)

[Popis hlavních souborů 8](#_Toc188883722)

[**Funkcionalita podle požadavků** 9](#_Toc188883723)

[Splněné požadavky 9](#_Toc188883724)

[Splnění technických požadavků 10](#_Toc188883725)

[Závěr 10](#_Toc188883726)

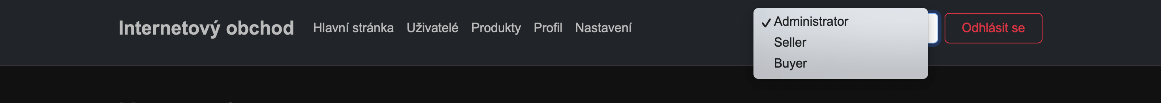
## **Popis aplikace**

Aplikace je interaktivní front-end webová aplikace vytvořená v Reactu, která poskytuje uživatelské rozhraní pro správu internetového obchodu. Internetový obchod obsahuje tři role uživatelů: Administrátor, Prodejce a Zákazník. Data jsou načítána z lokálního JSON serveru a aplikace umožňuje případnou budoucí integraci s back-end API.

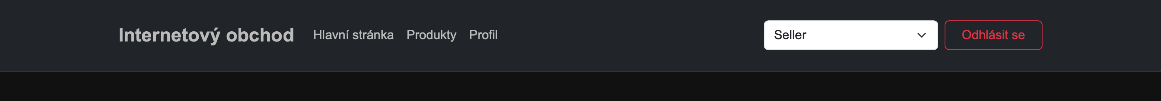
### Funkce aplikace

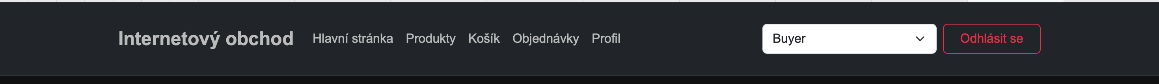
1. **Přihlašování uživatele**:
   * Jednoduchý přihlašovací formulář bez registrace.
   * Na základě přihlašovacích údajů je uživateli přiřazena role (Administrátor, Prodejce, Zákazník).

(Administrátor):



(Prodejce):



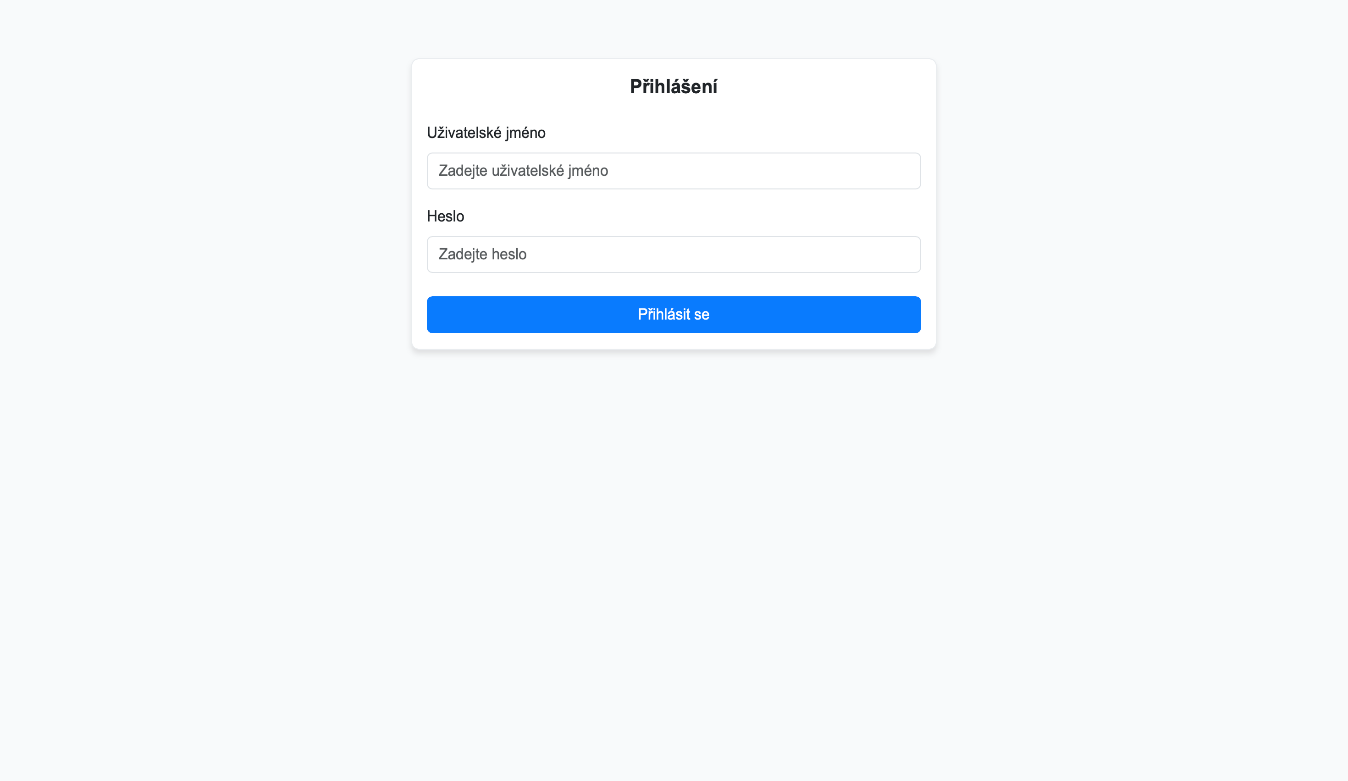
(Zákazník):  


* + Role ovlivňují přístup k jednotlivým funkcím aplikace.

1. **Správa rolí a uživatelského rozhraní**:
   * Uživatelé mohou přepínat mezi rolemi, pokud mají přístup k více rolím.
   * Layout aplikace se dynamicky mění na základě aktuální role.
   * **Individuální menu pro každou roli:**
     + **Administrátor**:
       - Přístup k přehledu všech uživatelů.
       - Možnost přidávat, upravovat nebo mazat uživatele.
       - Správa produktů a přístup k nastavení aplikace.
     + **Prodejce**:
       - Možnost přidávat a upravovat produkty.
       - Přehled dostupných produktů.
       - Omezený přístup k uživatelským datům.
     + **Zákazník**:
       - Přístup k produktům, možnost přidávat produkty do košíku.
       - Historie objednávek.
2. **Interakce s daty**:
   * Aplikace načítá a zpracovává data o uživatelích, produktech a objednávkách z db.json souboru.
   * Uživatelé mohou filtrovat a upravovat data v závislosti na svých oprávněních.
3. **Interaktivní komponenty**:
   * Filtrování produktů podle kategorií.
   * Modální okna pro detailní náhled nebo úpravu uživatelských dat.
   * Editační formuláře pro změnu obsahu, které aktualizují data v JSON souboru.
4. **Podpora pro rozdílné role**:
   * **Administrátor**: Přístup ke správě uživatelů, produktů a nastavení.
   * **Prodejce**: Možnost správy vlastních produktů.
   * **Zákazník**: Přístup k nákupu, košíku a historii objednávek.

### Obrazovky aplikace

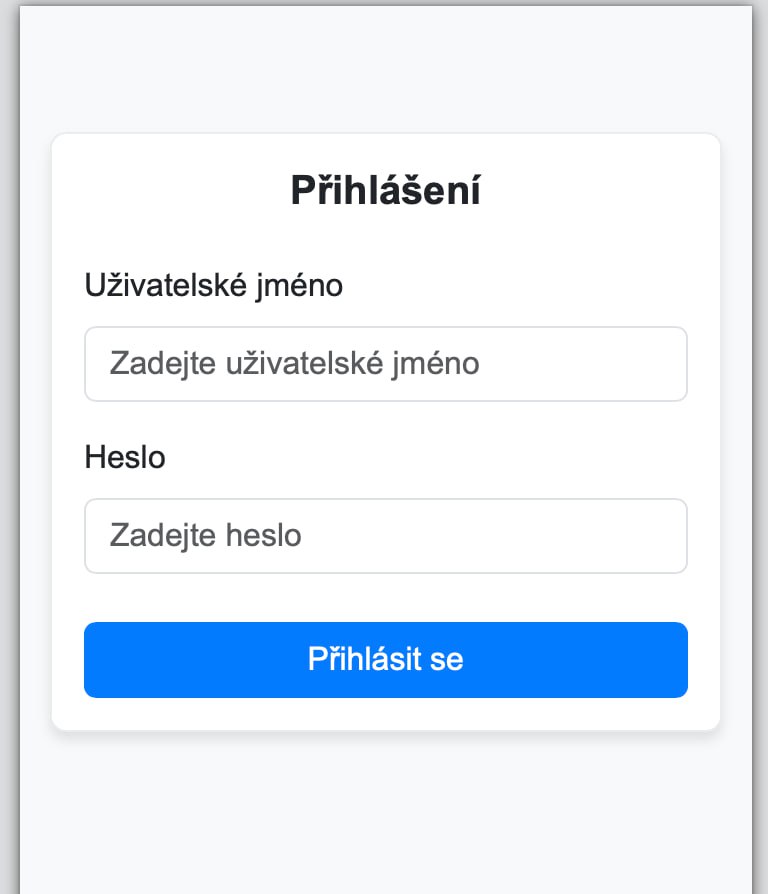
1. **Přihlašovací stránka**
   * Formulář pro přihlášení uživatele:

****

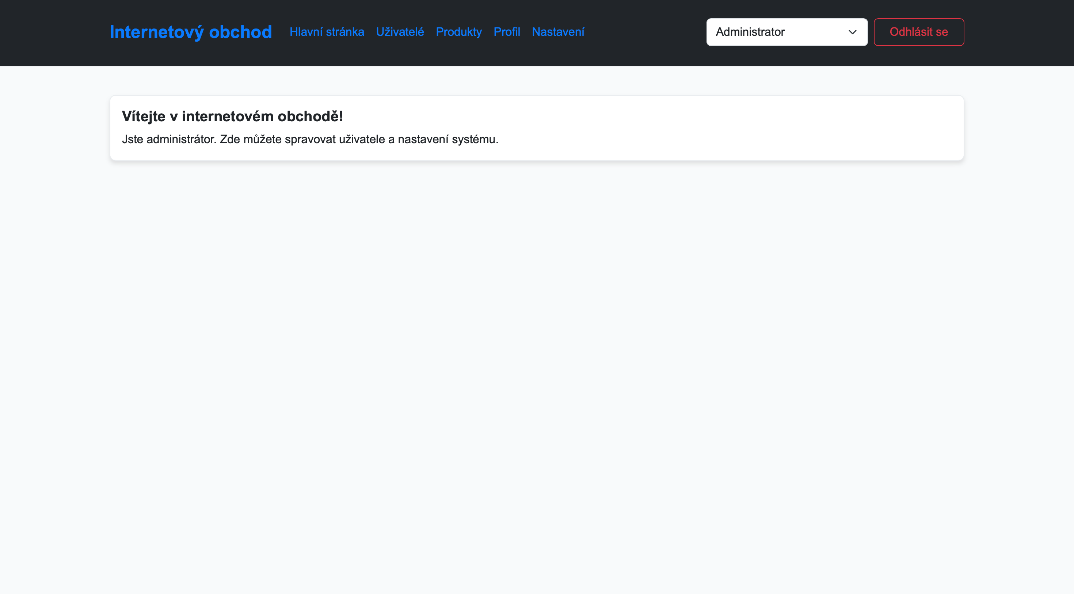
Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

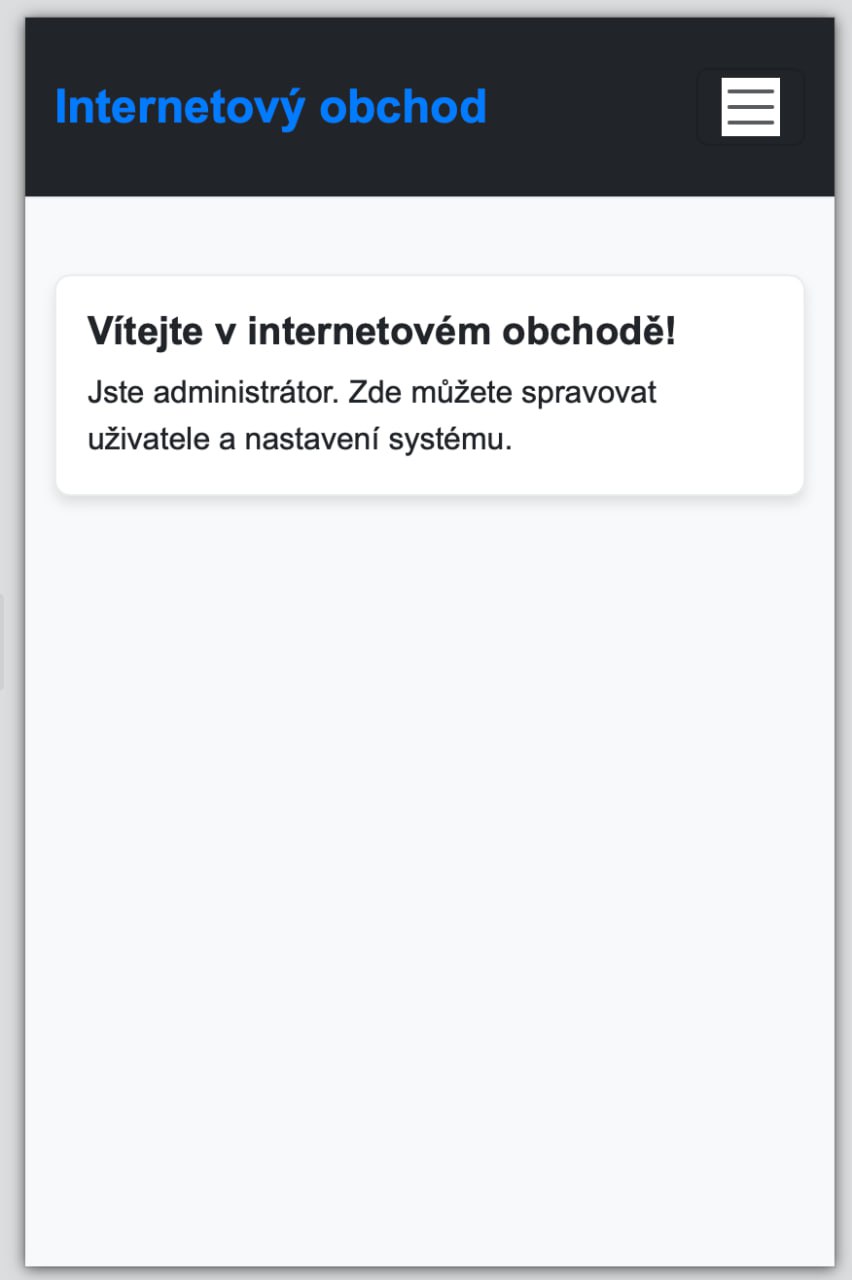
* + Pro telefon:

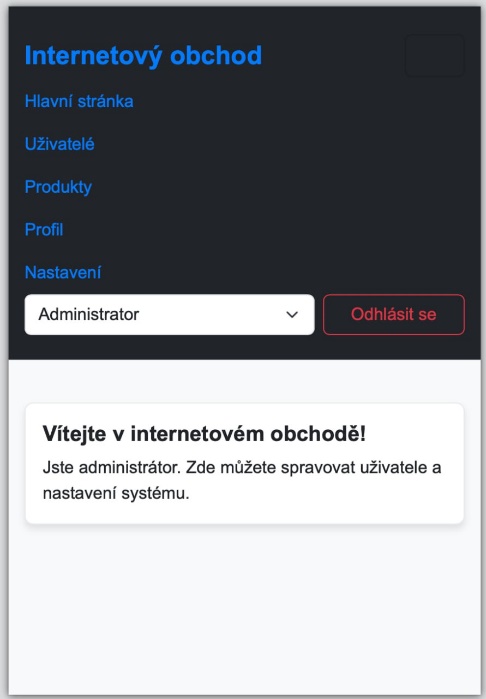


1. **Hlavní přehled (Dashboard)**
   * Zobrazení personalizovaných informací podle role:

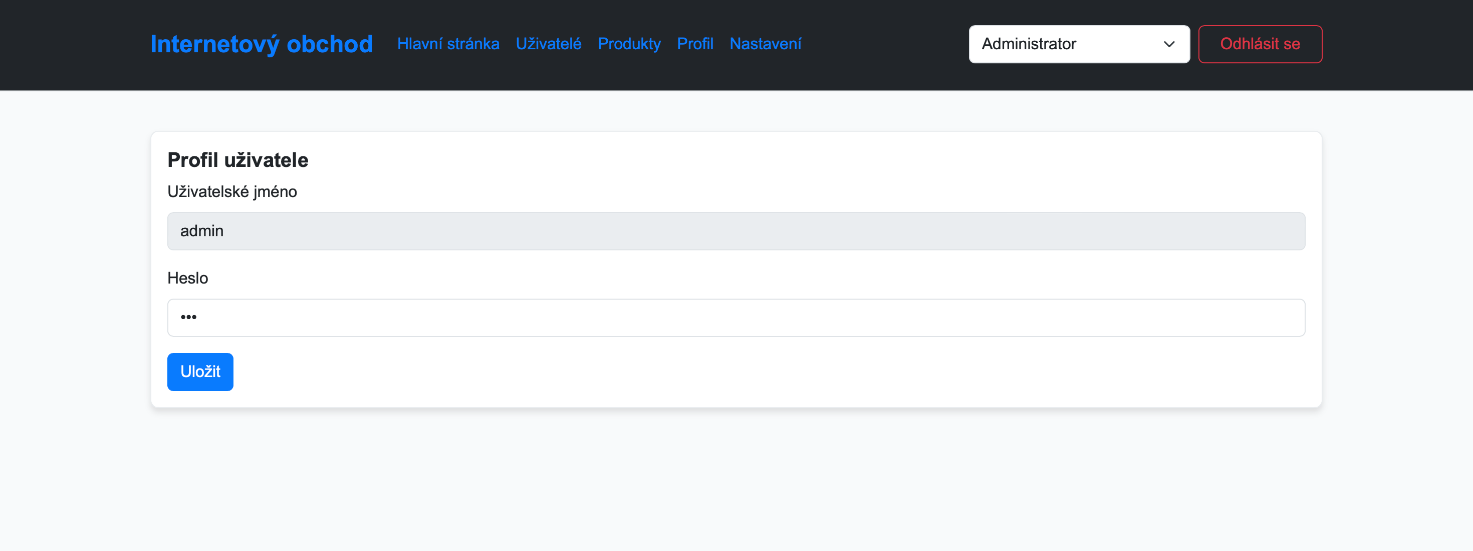


* + Pro telefon:

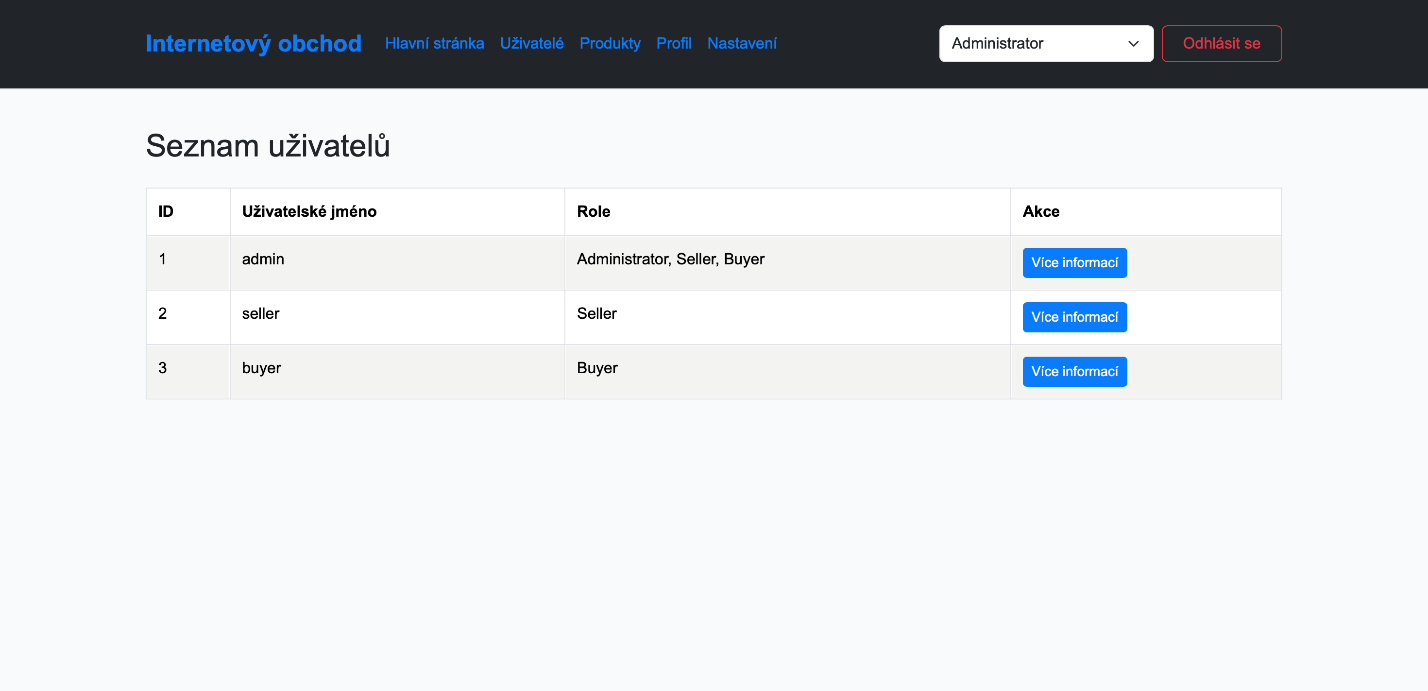


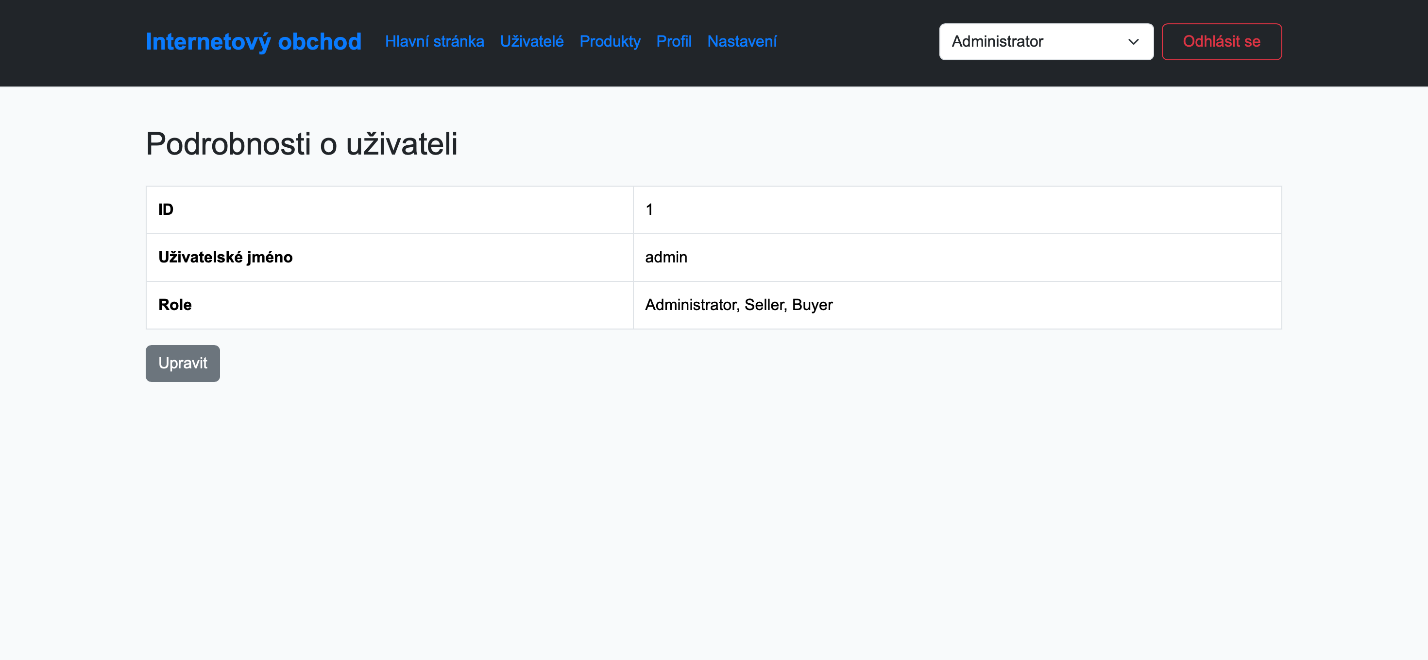


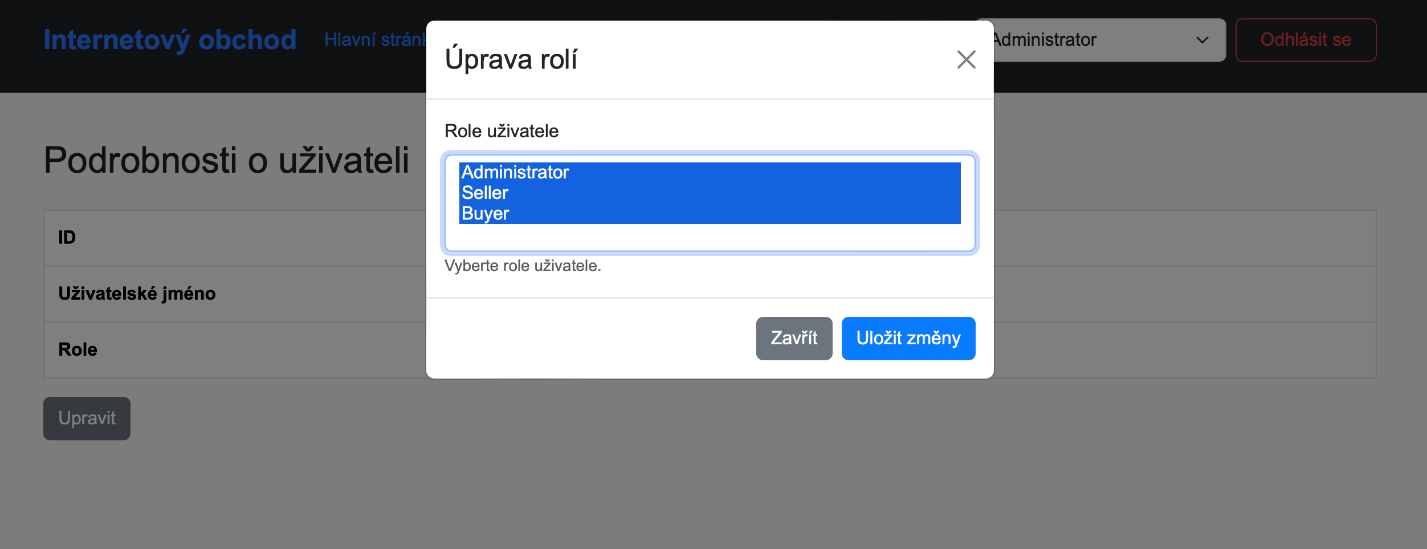
1. **Profil uživatele**
   * Změna hesla:



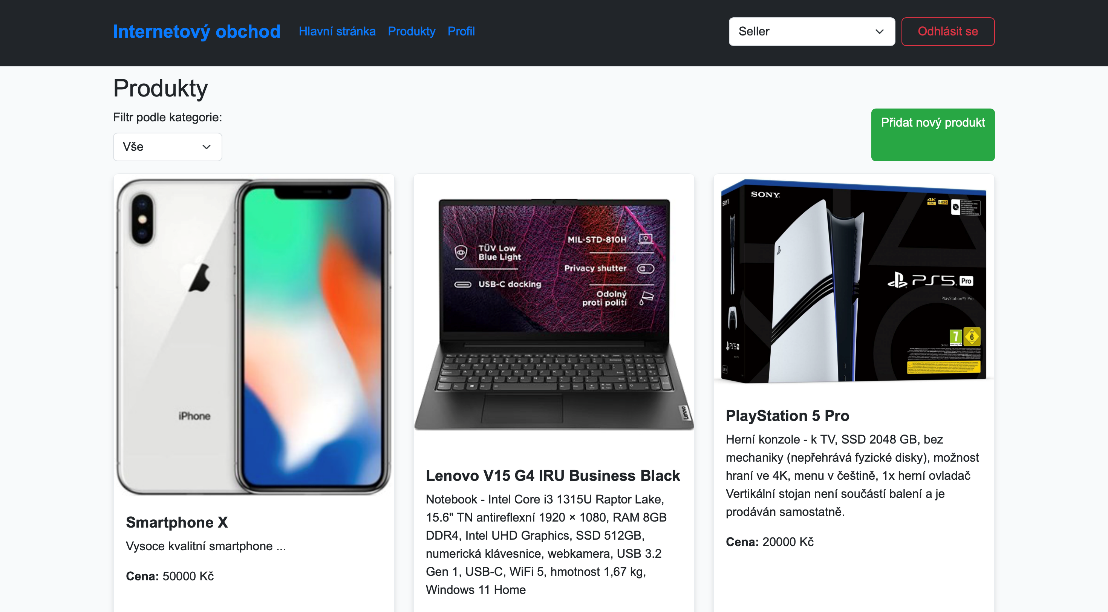
1. **Správa uživatelů (pouze pro Administrátory)**
   * Zobrazení seznamu uživatelů s možností detailní úpravy rolí:

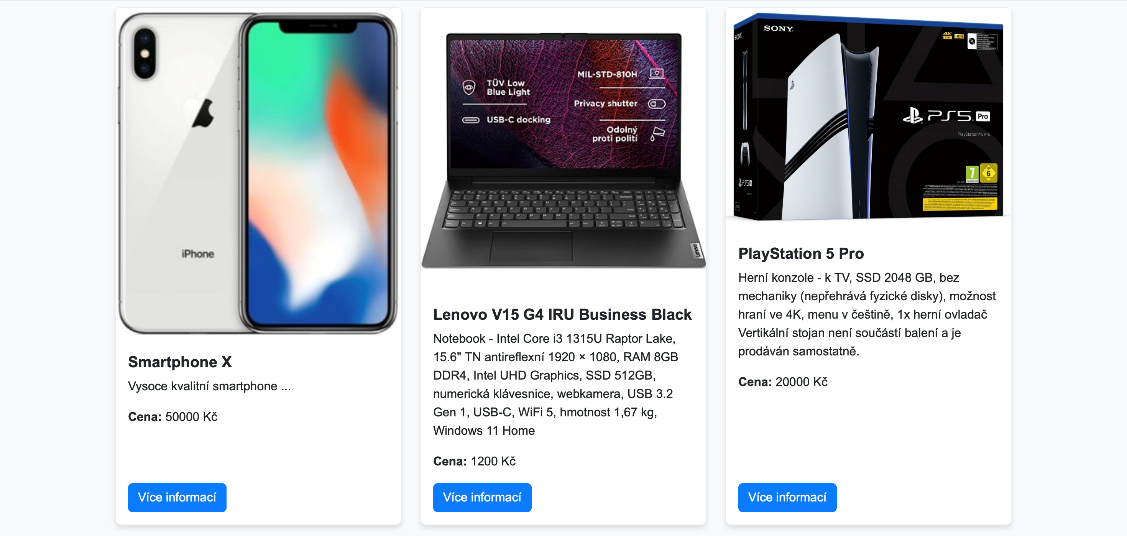




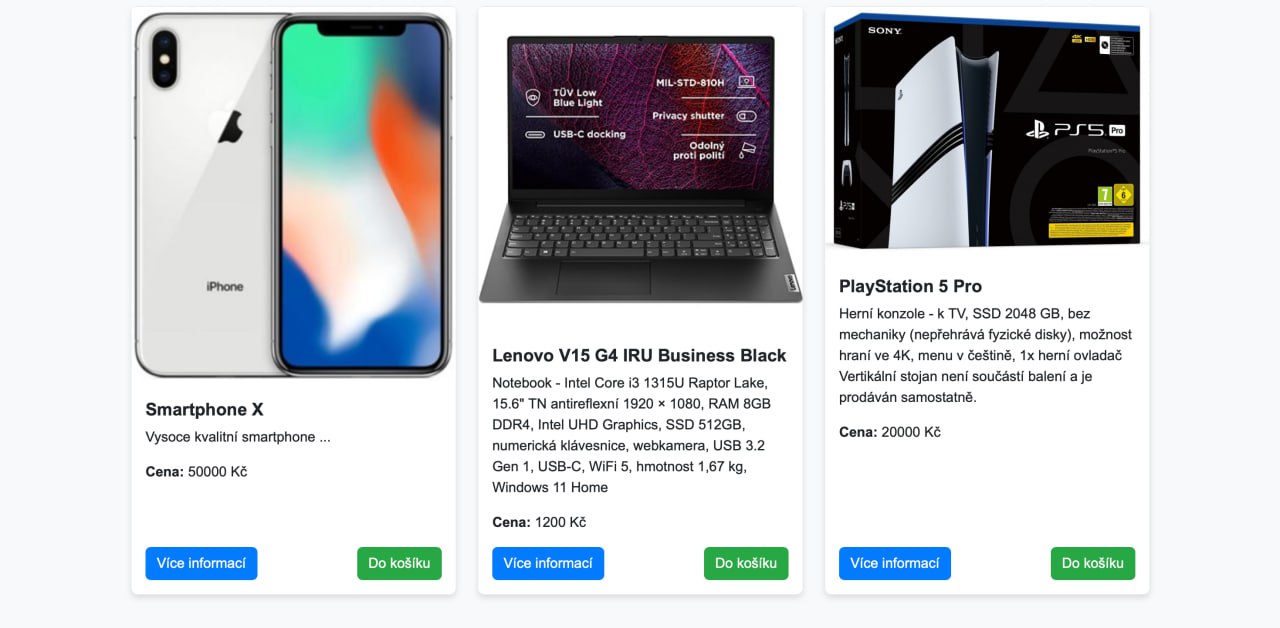


1. **Seznam produktů**
   * Filtrování a zobrazení produktů podle kategorií:

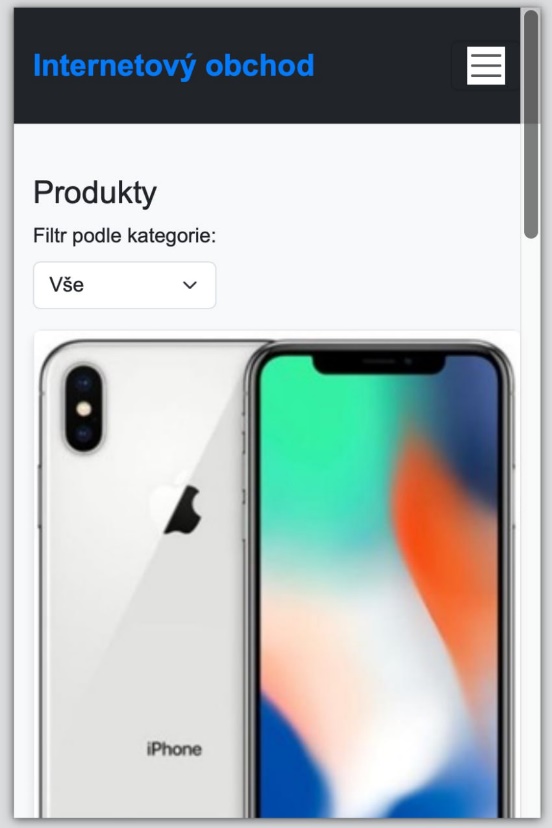




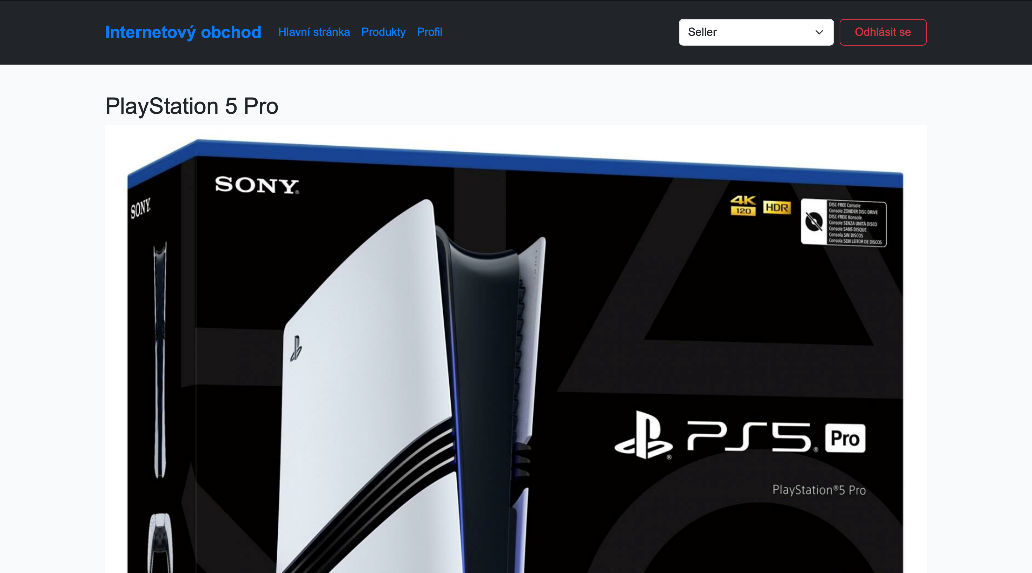
* + (dle role) „Buyer“:

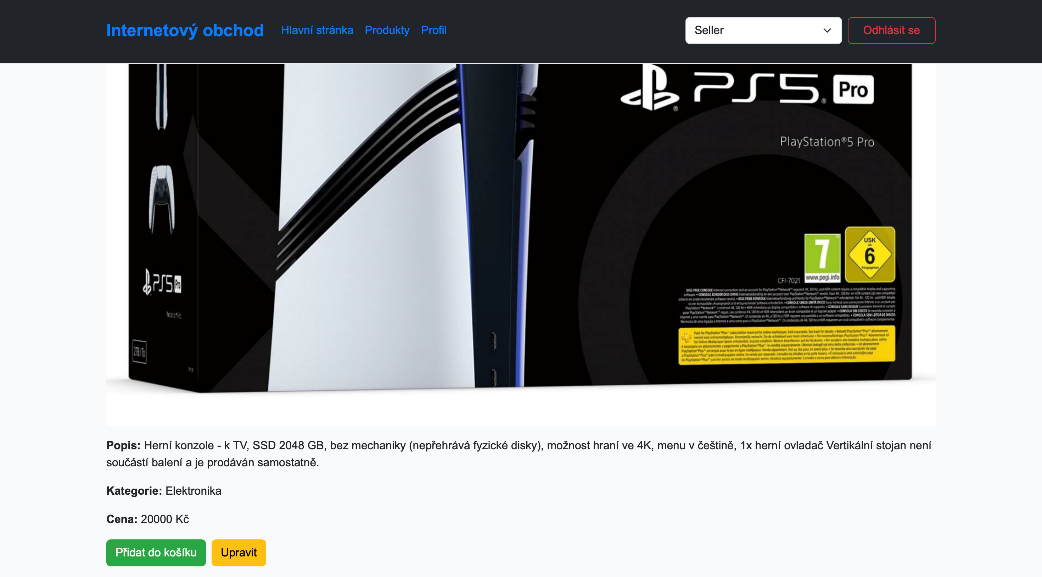


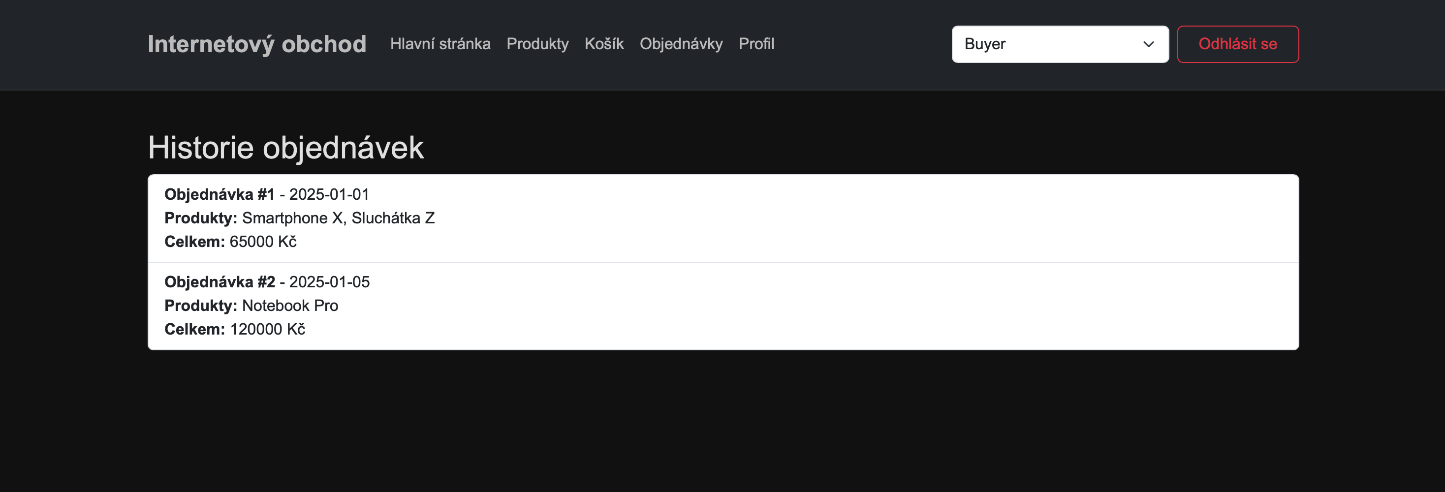
* + Pro telefon:

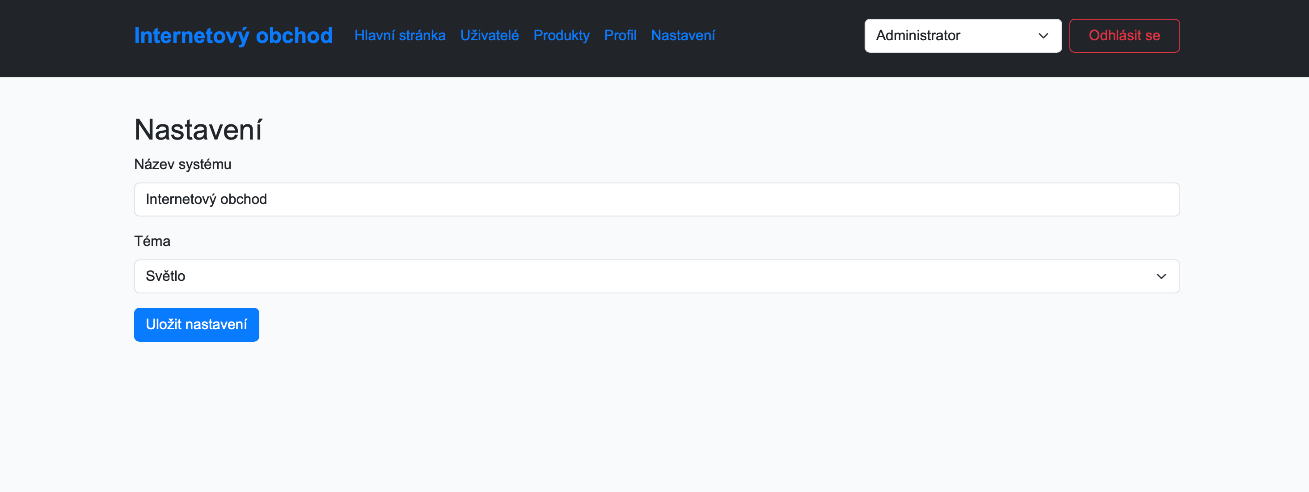


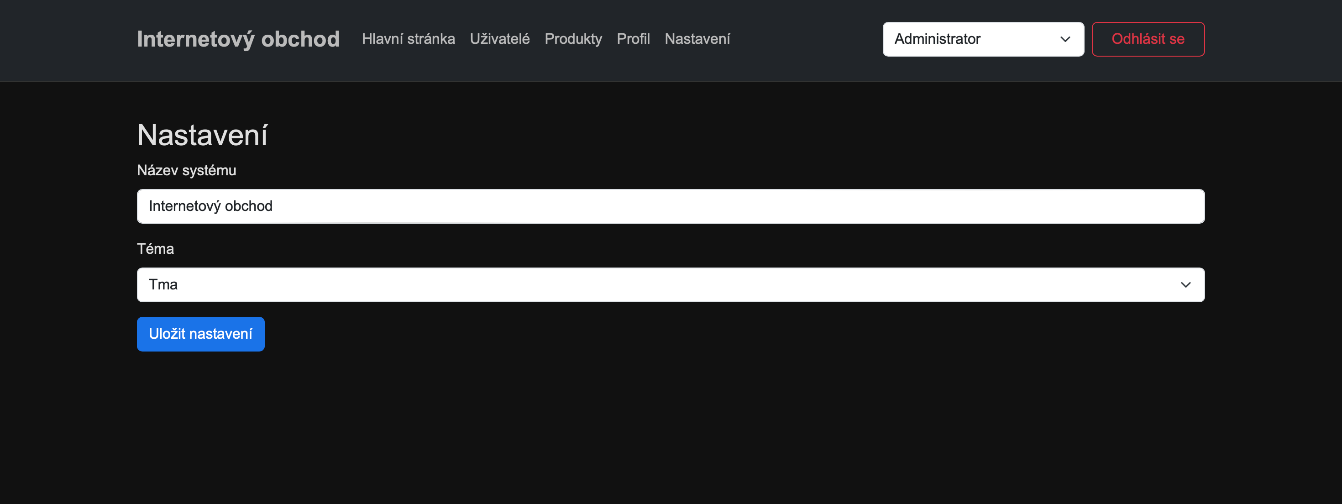
1. **Detail produktu**
   * Zobrazení podrobných informací o produktu s možností úpravy (dle role):





1. **Objednávky (pouze pro Zákazníky)**
   * Historie objednávek s detaily:  
     
2. **Nastavení (pouze pro Administrátory)**
   * Konfigurace aplikace:





## **Technický popis**

### Použité technologie

* **React.js**: Framework pro tvorbu uživatelských rozhraní.
* **React Bootstrap**: Pro stylizaci a zajištění responzivního designu.
* **json-server**: Pro simulaci back-end API.
* **Fetch API**: Pro komunikaci s json-serverem.

### Popis hlavních souborů

1. **App.jsx**:
   * Hlavní komponenta aplikace, která definuje navigaci a různé role.
2. **Navbar.jsx**:
   * Navigační panel, který se mění na základě aktuální role uživatele.
3. **Login.jsx**:
   * Zajišťuje přihlášení uživatele pomocí validace oproti db.json.
4. **ProductsList.jsx**:
   * Zobrazuje seznam produktů s možností filtrování a úpravy.
5. **Profile.jsx**:
   * Umožňuje uživateli změnit heslo a aktualizovat ho v db.json.
6. **json-server**:
   * Slouží jako simulovaný back-end. Data jsou uložena v db.json a upravují se pomocí HTTP metod.

## **Funkcionalita podle požadavků**

### Splněné požadavky

1. **Přihlašování uživatele**

* Jednoduchý formulář.
* Validace uživatelských údajů oproti JSON.

2. **Správa rolí**

* Uživatelé mohou přepínat role (pokud mají více než jednu roli) pomocí rozbalovacího menu v navigačním panelu.
* Po přepnutí role se automaticky aktualizuje uživatelské rozhraní (dashboard, dostupné funkce, a další komponenty).
* Role a jejich specifické možnosti:
  + **Administrátor**: Přístup ke všem funkcím (správa uživatelů, produktů, nastavení).
  + **Prodejce**: Možnost přidávat nové produkty, upravovat své produkty.
  + **Zákazník**: Přístup k produktům, košíku, historii objednávek.

1. **Interakce s daty**

* Použití json-server pro simulaci API a práci s daty (CRUD operace na uživatelích, produktech a objednávkách).
* Data jsou dynamicky načítána a upravována na základě akcí uživatelů:
  + Administrátor může spravovat seznam uživatelů (měnit role, upravovat údaje).
  + Prodejce může spravovat produkty (přidávat nové, upravovat existující).
  + Zákazník může prohlížet produkty, přidávat je do košíku a vytvářet objednávky.

1. **Interaktivní komponenty a vazby mezi daty**

* **Filtrování**: Umožňuje uživatelům filtrovat produkty podle kategorií.
* **Modální okna**: Používají se pro detailní náhled na uživatelská data nebo produkty.
* **Formuláře**: Umožňují úpravy uživatelských a produktových dat, která se okamžitě aktualizují v db.json.
* **Vazby mezi daty**: Například propojení objednávek se zákazníky nebo produktů s kategoriemi.

1. **Minimální počet obrazovek**

Aplikace obsahuje více než osm unikátních obrazovek, splňuje tedy požadavek na různorodost rozhraní.

### Splnění technických požadavků

* **Moderní technologie**: React.js, Bootstrap, json-server.
* **Dynamická práce s daty**: Fetch API a HTTP metody pro komunikaci s json-server.
* **Modularita**: Aplikace je rozdělena do přehledných komponent.
* **Responzivita**: Bootstrap zajišťuje přizpůsobení rozhraní na různých zařízeních.
* **Správa stavu uživatele**: React useState pro sledování aktuálního přihlášeného uživatele a jeho role.

## Závěr

Aplikace splňuje všechny požadavky zadání:

* Poskytuje přehledné uživatelské rozhraní s dynamickým přizpůsobením na základě role uživatele.
* Implementuje všechny potřebné funkce (přihlašování, správa dat, přepínání rolí).
* Používá moderní front-end technologie a podporuje budoucí rozšíření (např. propojení s reálným back-endem).
* Splňuje požadavky na minimální počet obrazovek a interaktivitu.