

RUNIX E-SHOP – DOKUMENTACE

RUNIX je moderní webová aplikace pro online nakupování oblečení a obuvi, postavená na technologickém stacku React (frontend), Express (backend), PostgreSQL (databáze) a Stripe (platby). Cílem projektu je ukázat kompletní e-shop řešení s uživatelským účtem, košíkem, objednávkami a bezpečnými online platbami.

1. POUŽITÉ TECHNOLOGIE A ARCHITEKTURA

FRONTEND

- React – komponentově orientovaný frontend framework.
- react-router-dom – pro směrování a více stránek ve SPA.
- CSS / SCSS – pro stylování komponent.
- Axios – pro komunikaci s API.

BACKEND

- Node.js + Express – REST API server.
- pg – PostgreSQL client pro Node.js.
- dotenv – správa konfigurací.
- Stripe – zpracování online plateb.

DATABÁZE

PostgreSQL – relační databáze pro ukládání produktů, uživatelů, objednávek.

ARCHITEKTURA

Projekt je rozdělen na dva samostatné servery:

- /client – React frontend běžící na `http://localhost:3000`.
- /server – Express backend běžící na `http://localhost:5000`.

Oba servery spolu komunikují přes REST API. Data jsou uložena v PostgreSQL databázi.

2. INSTALACE A SPUŠTĚNÍ

PŘEDPOKLADY

- Nainstalovaný Node.js a npm.
- Nainstalovaný PostgreSQL server.

POSTUP

2.1 KLONOVÁNÍ REPOZITÁŘE

```
-git clone https://github.com/tvoje-jmeno/runix-shop.git  
-cd runix-shop
```

2.2 INSTALACE ZÁVISLOSTÍ

Frontend:

```
cd client  
npm install
```

Backend:

```
cd ../server  
npm install
```

2.3 VYTVOŘENÍ DATABÁZE

```
-Přihlaste se do PostgreSQL a vytvořte databázi:  
-CREATE DATABASE products;
```

2.4 IMPORT DATABÁZOVÝCH SCHÉMAT A DAT

```
psql -d runix -f db/schema.sql  
psql -d runix -f db/seed.sql
```

```
- schema.sql – vytvoří potřebné tabulky.  
- seed.sql – vloží ukázkové produkty a uživatele.
```

2.5 SPUŠTĚNÍ SERVERŮ

Backend:

```
cd server  
npm run dev
```

Frontend:

```
cd ../client  
npm start
```

Přístup:

```
- Frontend: http://localhost:3000  
- Backend API: http://localhost:5000/api
```

3. STRUKTURA PROJEKTU

```
/client
/src
  /Components
    Header.jsx
    AutoLogin.jsx
    ...
  App.jsx
  index.js

/server
  /controllers
  /routes
  /models
  server.js
  db.js

/db
  schema.sql
  seed.sql
```

4. FRONTEND (REACT)

Frontend je postaven jako SPA pomocí Reactu. Používá react-router-dom pro směrování podle URL a obsahuje jednotlivé kategorie obchodu (ženy, muži, děti, nové, slevy, detail produktu, košík, platba...).

SOUBOR APP.JS

Soubor App.jsx definuje základní rozvržení aplikace, načítání session a jednotlivé stránky podle URL. Klíčové části:

- Loader komponenta – zobrazuje úvodní animaci.
- Promo box – zobrazuje propagační okno při spuštění.
- AutoLogin – automatické přihlášení uživatele ze session.
- Header – navigace mezi sekcemi.
- Routes – definují jednotlivé stránky aplikace.

5. BACKEND (EXPRESS + POSTGRESQL)

Backend aplikace tvoří REST API server postavený na Node.js a Express. Je zodpovědný za zpracování požadavků od React frontendu, správu autentizace uživatelů, práci s databází PostgreSQL a integraci platební brány Stripe.

HLAVNÍ KOMPONENTY BACKENDU:

- **Express** – rychlý a minimalistický webový server, poskytující API endpointy.
- **PostgreSQL** – relační databáze pro ukládání produktů, uživatelů a objednávek. K připojení k databázi se používá balíček pg.
- **Autentizace** – JWT tokeny a session middleware (express-session) zajišťují bezpečné přihlášení a autorizaci uživatelů.
- **Bezpečnost** – ochranné hlavičky pomocí helmet, omezení rychlosti pomocí express-rate-limit, ochrana proti XSS a CSRF.

6. PLATBY (STRIPE)

Pro platby je použit Stripe Checkout, který poskytuje bezpečné platební rozhraní. Platby jsou zpracovány výhradně na backendu — frontend odešle položky k zaplacení a backend vytvoří platební session na serveru Stripe.

PROCES PLATBY:

- Frontend odešle POST požadavek na endpoint /createCheckout s položkami k zaplacení.
- Server zavolá Stripe API (stripe.checkout.sessions.create) a vytvoří Checkout session.
- Server vrátí URL na Stripe Checkout, kam je uživatel přesměrován.
- Po úspěšné platbě je uživatel přesměrován zpět na /success nebo /cancel.