



Gymnázium, Praha 6, Arabská
předmět Webové aplikace

E-shop

Ročníkový projekt

Anotace

Mým ročníkovým projektem je vytvoření webového e-shopu. E-shop jsem vytvořil v React.js a Node.js. Front-end je vytvořen v React.js, back-end běží na Node.js a jako databázi jsem použil MongoDB.

Zadání projektu

Vytvořit e-shop v React.js a Node.js.

Obsah

1. Úvod	4
2. Struktura	4
3. Front-end	4
3.1 Nabídka produktů	4
3.2 Informace o produktu	5
3.3 Košík	5
3.4 Admin login	6
3.5 Správa produktů	6
3.6 Přidávání a úprava produktů	6
4. Back-end	7
4.1 Modely	7
4.2 Middleware	7
4.3 Http cesty	7
5. Databáze	8
6. Jak projekt spustit	8
7. Závěr	8
8. Seznam obrázků	8
9. Zdroje	8

1. Úvod

Toto téma jsem si zvolil, protože mi přišel e-shop jako užitečný nápad. Práci jsem se rozhodl udělat v React.js a Node.js, protože jsme se měli ve škole učit práci s touto knihovnou a vývojovým prostředím.

2. Struktura

Projekt je složen ze tří částí: front-endu, back-endu a databáze. Front-end, zjednodušeně řečeno je to, co uživatel uvidí na obrazovce monitoru, je vytvořen pomocí knihovny React.js. Back-end, tedy část programu, která interaguje s databází, byl vytvořen v Node.js. Jako databázi jsem si zvolil MongoDB, která je NoSQL databází.

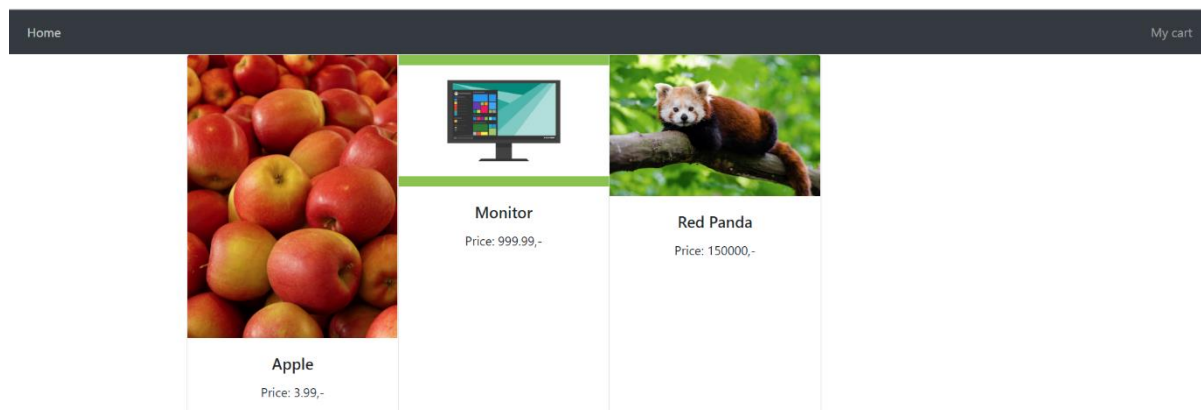
3. Front-end

Front-end se skládá ze dvou celků. Prvním jsou stránky přístupné pro všechny uživatele, který se nepotřebuje přihlásit, a druhým celkem je skupina stránek přístupná pro administrátora e-shopu. Uživatelům přístupnými stránkami jsou stránka s nabídkou produktů, detailnější stránky o jednotlivých produktech a košík s produkty. Použitými knihovnami jsou Bootstrap, Redux, Axios a React-router-dom. Bootstrap zlehčuje úpravu vzhledu stránky. Redux umožňuje mít globální proměnné. Knihovna Axios zjednodušuje posílání http požadavků a React-router-dom je pro navigaci mezi jednotlivými stránkami.

3.1 Nabídka produktů

Na této stránce uživatel vidí všechny dostupné produkty v e-shopu. Produkty jsou zobrazeny v podobě kartiček s obrázkem a základními informacemi o produktu. Po kliknutí na libovolný produkt je uživatel přesměrován na stránku s podrobnějšími informacemi o produktu.

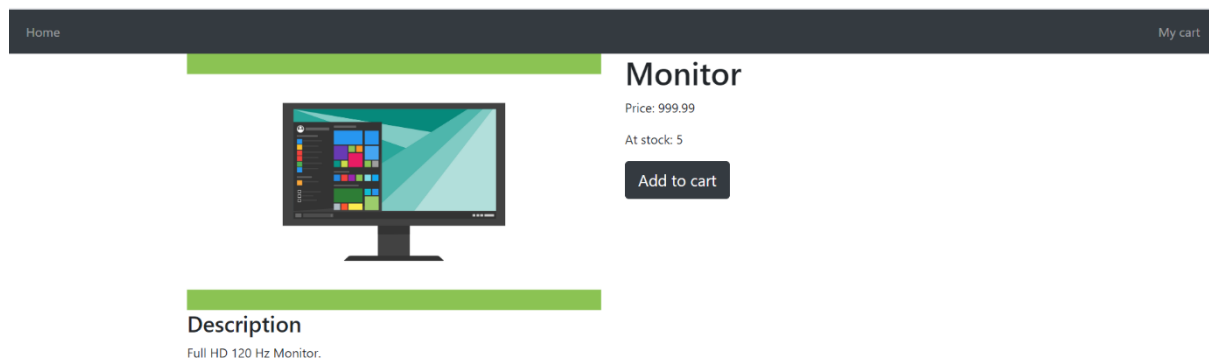
Produkty jsou načteny z databáze přes back-end a jsou uloženy do Redux store, což je v podstatě globální správce proměnných.



Obrázek 1 - Home page

3.2 Informace o produktu

Zde může návštěvník vidět nezmenšený obrázek s produktem a opět základní informace o produktu. Kromě toho je zde i detailní popis produktu. Také je tam tlačítko, které po kliknutí přidá produkt do košíku.

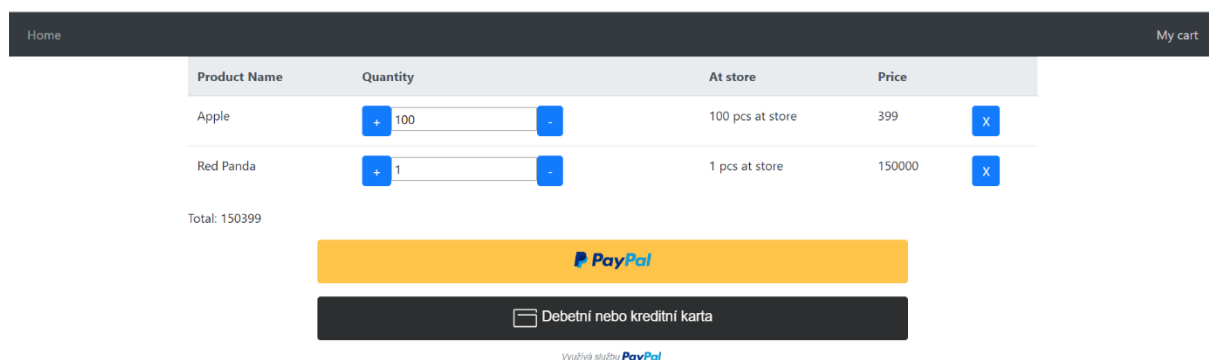


Obrázek 2 - Description page

3.3 Košík

Na stránce košíku jsou zobrazeny produkty, které si do něj uživatel vložil. Uživatel může produkty odebrat, či změnit jejich množství v košíku. Když je v košíku alespoň jeden produkt, tak se uživateli zobrazí tlačítko PayPal, pomocí kterého může zaplatit (zatím jsou transakce jen v rámci PayPal sandboxu). Po uživatelově nákupu dostane prodávající upozornění od PayPalu se seznamem nakoupených produktů. Kód k uskutečnění platební brány PayPal byl převzat z (fnky & codediodeio, 2020) a mírně upraven.

Košík je součástí Redux store, to znamená, že jsem mohl jeho obsah upravovat i z jiných stránek než z něj. Je možné z něho odstraňovat produkty nebo nastavit množství k zakoupení.



Obrázek 3 - Shopping cart

3.4 Admin login

Když není admin přihlášen, tak je při navštívení stránky vyžadující přihlášení přesměrován na tuto stránku. Zde se vyplní uživatelské jméno a heslo a je po úspěšném přihlášení přesměrován na stránku se správou produktů. Admina je možné vytvořit manuálním přidáním admina v databázi do kolekce „users“ nebo prvního admina je možné také přidat http požadavkem. Kód vztahující se k přihlašování byl převážně napsán podle zdrojů patřících (Traversy Media, 2020).

Welcome

Username

Password

Login

Obrázek 4 - Admin login page

3.5 Správa produktů

Na této stránce má uživatel možnost upravovat, smazat produkt nebo vytvořit nový produkt. Po kliknutí na tlačítko na vytvoření produktu je uživatel přesměrován na stránku s formulářem pro přidání produktu. Pro úpravu produktu je uživatel přesměrován na stránku, kde produkt může upravit a kliknutím na tlačítko se znakem „X“ produkt smaže.

Product Name	Quantity	Price	Create New Product	
Apple	100 pcs at store	3.99	Edit	Delete
Monitor	5 pcs at store	999.99	Edit	Delete
Red Panda	1 pcs at store	150000	Edit	Delete

Obrázek 5 - Manage products

3.6 Přidávání a úprava produktů

Na těchto stránkách uživatel může vyplnit formuláře nebo upravit jejich obsah pode toho, jestli vytváří nový produkt nebo jej upravuje. Pro vytvoření produktu je nutné zadat název, cenu, počet kusů na skladu, popis a nahrát obrázek produktu. Metoda pro nahrávání obrázku byla převzata z (bradtraversy, 2020).

Create New Product

Name:

Price

At Store

Description

Image

 Soubor nevybrán

Obrázek 6 - Add product

4. Back-end

Jak již bylo zmíněno back-end běží na Node.js. Kromě toho používám framework Express a Mongoose, který zjednodušuje práci s MongoDB databází. Dále používám knihovny Bcrypt, a JWT. Bcrypt a JWT používám pro bezpečné přihlašování admina. Také přijímá http požadavky z front-endu a poté podle nich provádí operace v databázi. Například vrátí seznam produktů, vloží nový produkt do databáze atd.

4.1 Modely

Díky frameworku Mongoose můžu definovat, jaké vlastnosti mají mít objekty, které se uloží v databázi. Moje webová stránka potřebuje pouze dva modely; jeden na ukládání produktů v databázi a druhý na ukládání přihlašovacích údajů admina.

4.2 Middleware

Pro zvýšení bezpečnosti používám middleware převzatý z návodů na stránce YouTube, který ověřuje identitu odesílatele požadavků (Traversy Media, 2020). Tento middleware je použit u všech příkazů, ke kterým má přístup pouze admin. Mezi takové požadavky patří přidávání a upravování produktů nebo jejich odstranění. Middleware byl převzat ze zdrojů patřících (Traversy Media, 2020).

4.3 Http cesty

Když do back-endu přijde požadavek, tak po případném ověření odesílatele je proveden jeden nebo více příkazů v databázi. Pro přehlednost jsou požadavky roztříděny do tří kategorií a to podle http adresy. Produkty mají svoji vlastní cestu, uživatel a autentifikace také. Cesty pro produkty obsahují např. přečtení všech produktů z databáze či základní operace s nimi. Uživatelské požadavky můžou vytvořit nového admin, ale jen za určitých okolností nebo přečíst o něm informace. Autentifikační cesta slouží k přihlášení admina a je založena na kódu ze zdrojů patřících (Traversy Media, 2020).

5. Databáze

Používám NoSQL MongoDB databázi, která je hostována na serverech MongoDB Atlas, protože je to bezplatné. Tento typ databáze jsem si zvolil, jelikož je často používána v kombinaci s Express a React.js.

6. Jak projekt spustit

Požadováno je runtime prostředí Node.js. Pro úplnou funkčnost programu je potřeba v souboru .env do proměnné „ATLAS_URI“ napsat URI databáze a v \client\public\index.html ve skriptu na čtyřiačtyřicátém řádku nahradit „SB_ID“ client ID sandbox aplikace. Dále je potřeba nainstalovat všechny potřebné moduly pomocí příkazu „npm install“. Server se poté spustí příkazem „yarn start“ a klient pomocí příkazu „npm start“ zadaným ve složce \client.

7. Závěr

Poprvé jsem pracoval s knihovnou React.js a prostředím Node.js. Při programování této práce jsem se naučil mnoho nových a užitečných technologií. E-shop obsahuje i stránky pro administrátora a platební bránu PayPal. Konečný program se mi bezproblémově podařilo spustit na mém počítači.

8. Seznam obrázků

Obrázek 1 - Home page	4
Obrázek 2 - Description page	5
Obrázek 3 - Shopping cart	5
Obrázek 4 - Admin login page	6
Obrázek 5 - Manage products	6
Obrázek 6 - Add product	7

9. Zdroje

Axios. (21. Květen 2020). Načteno z Npmjs: <https://www.npmjs.com/package/axios>

Bcrypt.js. (21. Květen 2020). Načteno z Npmjs: <https://www.npmjs.com/package/bcryptjs>

bradtraversy. (21. Květen 2020). *react_file_uploader*. Načteno z GitHub:

https://github.com/bradtraversy/react_file_uploader/blob/master/server.js

Classic-home. (21. Květen 2020). Načteno z PayPal: <https://developer.paypal.com/classic-home/>

fnky, & codediodeio. (21. Květen 2020). *193-paypal-checkout-v2-demos*. Načteno z GitHub: <https://github.com/fireship-io/193-paypal-checkout-v2-demos/blob/master/react-app/src/App.js>

Home. (21. Květen 2020). Načteno z Node.js: <https://nodejs.org/>

Jsonwebtoken. (21. Květen 2020). Načteno z Npmjs:
<https://www.npmjs.com/package/jsonwebtoken>

MongoDB Atlas. (21. Květen 2020). Načteno z MongoDB: <https://www.mongodb.com/>

Mongoose. (21. KVěten 2020). Načteno z Npmjs:
<https://www.npmjs.com/package/mongoose>

React. (21. Květen 2020). Načteno z React.js: <https://reactjs.org/>

React Bootstrap. (21. Květen 2020). Načteno z GitHub: <https://react-bootstrap.github.io/>

React Redux. (21. Květen 2020). Načteno z React Redux: <https://react-redux.js.org/>

React-router-dom. (21. Květen 2020). Načteno z Npmjs:
<https://www.npmjs.com/package/react-router-dom>

Traversy Media. (21. Květen 2020). *Learn The Mern Stack [10] - React & Redux Auth State.*
Načteno z YouTube: <https://www.youtube.com/watch?v=qyomEaXQJFk>

Traversy Media. (21. Květen 2020). *Learn The Mern Stack [11] - React Registration.* Načteno z
YouTube: https://www.youtube.com/watch?v=de5gkk_40Eo

Traversy Media. (21. Květen 2020). *Learn The Mern Stack [12] - React Login & Access Control.*
Načteno z YouTube: <https://www.youtube.com/watch?v=EfwP65XF0bM>

Traversy Media. (21. Květen 2020). *Learn The Mern Stack [9] - Backend JWT Auth.* Načteno z
YouTube: <https://www.youtube.com/watch?v=USaB1adUHM0>