 **Gymnázium, Praha 6, Arabská 14**  
Ročníkový projekt, obor Programování



# BookExchange

## Bazar Učebnic Arabská

Vypracoval:  
Vedoucí práce:

Josef Litoš  
Ing. Daniel Kahoun

únor 2022

*Prohlašuji, že jsem jediným autorem tohoto projektu, všechny citace jsou řádně označené a všechna použitá literatura a další zdroje jsou v práci uvedené. Tímto dle zákona 121/2000 Sb. (tzv. Autorský zákon) ve znění pozdějších předpisů uděluji bezúplatně škole Gymnázium, Praha 6, Arabská 14 oprávnění k výkonu práva na rozmnožování díla (§ 13) a práva na sdělování díla veřejnosti (§ 18) na dobu časově neomezenou a bez omezení územního rozsahu.*

V ..... dne .....

Podpis autora .....

**Anotace:** Tento projekt si klade za cíl vytvořit stránku, na které budou moci studenti inzerovat své učebnice, které již nepotřebují, a žáci nižších ročníků si je tak budou moci pořídit. O průběhu obchodu jsou uživatelé informováni prostřednictvím emailu, kterým se přes Google OAuth přihlásí. Na email dostanete všechna upozornění týkající se hledané knihy, příchozích požadavků, nebo změn některých z objednávek. Buď knihu dostanete Vy, nebo někdo jiný. V každém případě se o jejím prodání dozvíte.

**Abstract:** This project sets as a goal to create a page, on which students will be able to post their textbooks they are not in need of. Pupils of lower classes can then buy them. Users are informed about the current state of their trade through email that they will use in Google OAuth to log in. In fact, you will receive all notifications on your email being it a new book you were looking for, a new incoming request or a change in requests of your own. Either the book goes to you or to someone else. Both ways, you will know, when the book was sold.

**Abstrakt:** Dieses Projekt stellt sich vor, eine Seite zu machen, auf welcher Studenten ihre nicht mehr benutzte Lehrbücher anzeigen können. Schüler aus unteren Studienjahren können sich diese Anzeige ansehen und die Bücher kaufen. Benutzer werden über den Status des Handels per Email informiert. Die gleiche Email wird auch bei der Google OAuth Anmeldung benutzt. Ihre Email bekommt alle Hinweisen, soll das ein neues Buch, das Sie gesucht haben, neues Gesuch, oder eine Veränderung in Ihre Anträgen sein. Entweder geht das Buch zu Ihnen, oder zu jemandem anderen. In beiden Varianten werden Sie über seine Verkauf informiert.

**Klíčová slova / Keywords / Schlüsselwörter:** React, Node, SQL, Google OAuth2

**Zadání:** Vytvořte webovou aplikaci, která umožní uživateli následující

- reagovat na inzerovanou knihu
- zobrazit a filtrovat dostupné knihy
- přihlásit se o knihu, jakmile bude dostupná
- automatizované odstranění inzerátu po zakoupení
- přidat, odstranit a upravit vlastní inzeráty (včetně fotky)

**Bonusové funkce:**

- vlastní verze UI pro mobilní zařízení
- nabídka balíčku knih s výhodnější cenou
- administrátorský přístup do databází z webu
- integrace chatu pro domluvu na předání mezi uživateli

# Obsah

<b>Úvod</b>	<b>4</b>
<b>1 Databáze</b>	<b>5</b>
1.1 Uživatel . . . . .	5
1.2 Kniha . . . . .	6
1.3 Žádost . . . . .	6
1.4 Upozornění . . . . .	6
<b>2 Server</b>	<b>7</b>
2.1 Autentifikace . . . . .	7
2.2 Vstup do databáze . . . . .	7
2.3 Přístupové body . . . . .	7
2.3.1 /user . . . . .	8
2.3.2 /book . . . . .	8
2.3.3 /request . . . . .	9
2.3.4 /notify . . . . .	10
2.4 Zasílání emailů . . . . .	10
<b>3 Webová stránka</b>	<b>11</b>
3.1 Záhloví . . . . .	11
3.2 Kniha jako stavební jednotka . . . . .	11
3.3 Seznam knih . . . . .	11
3.4 Tvorba a úprava knih . . . . .	12
3.5 Management žádostí . . . . .	12
3.6 Překážky během vývoje . . . . .	12
<b>Závěr</b>	<b>13</b>
<b>Seznam použité literatury</b>	<b>14</b>
<b>Seznam obrázků</b>	<b>16</b>

# Úvod

Na začátek bych rád vysvětlil své myšlenkové pochody, kterými jsem byl doveden k tematiku této práce, a proč mi přijde tato aplikace vcelku užitečná.

Již po dva celé roky ovlivňuje probíhající pandemie fungování takřka všech státních i soukromnických aparátů. Kromě globálně známých problémů to způsobilo i mnoho lokálních potíží obzvlášť znatelných v dobách lockdownu. Příklady toho jsou častější rodinné spory, často zrušení sportovních kroužků, a pro tento text nejpodstatnější – zrušení konání burzy učebnic.

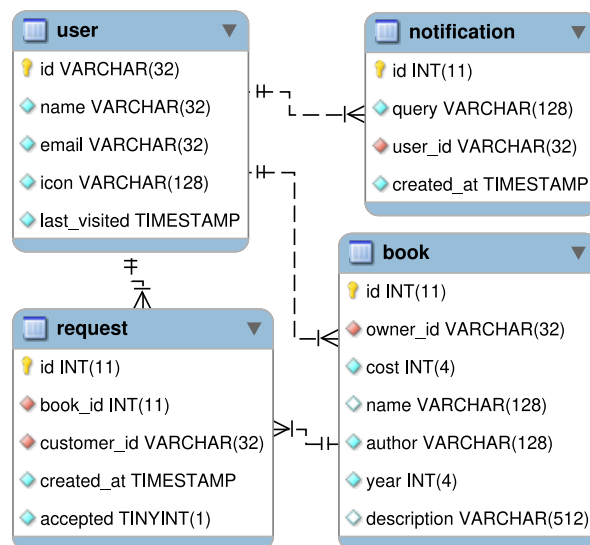
Jelikož z předešlých let se učebnice hromadily, a burza učebnic byla tou nejrychlejší, nejsnazší a často také nejvýhodnější cestou, jak učebnice vyřadit, jejich uskladňování nebylo ideálním řešením. V tu dobu ale žádná náhrada podobného stylu za školní burzu nebyla. Tím pádem se jevil jasný nápad – udělat si vlastní burzu a dát ji na stránky školy.

Dokumentace je rozdělena na tři celky uvedené v pořadí nástavby programu. Každý z celků projednává svou funkci v programu, vnitřní rozdělení, vzájemné vztahy a problémy s danou částí spojené.

# 1. Databáze

Chceme-li pracovat s daty a nějak je ukládat, je pravděpodobné, že se rozhodneme použít nějakou databázovou strukturu. V případě tohoto programu je využíváno prostředí SQL běžící na serveru MariaDB.

SQL má několik svých výhod a vhodných využití, pro které bylo zvoleno pro použití v tomto programu. Mezi ně patří vhodnost pro jednoduchá data (myšleno ne objekty), spolehlivost a uzpůsobení pro velká množství dat. Jiné databáze, jako třeba MongoDB, jsou vhodnější pro ukládání celých objektů, čehož tento projekt nevyužívá.



Obrázek 1.1: Diagram používané databáze

## 1.1 Uživatel

Tabulka uživatelů je naprosto základním prvkem, od kterého se veškeré dění dále odvíjí. Všechny ostatní tabulky se na ni odkazují. Pro vytvoření knihy musíme mít jasno v tom, kdo knihu přidal, a kdo o ní žádá. Každý uživatel by dále pravděpodobně rád viděl jasný znak, že je přihlášen, proto se ukládá i odkaz na jeho profilovou fotku, která se mu zobrazí po přihlášení do stránky.

V obrázku 1.1 v tabulce `user` můžeme vidět, že si program také udržuje přehled o posledním přihlášení uživatele, aby mohl v případě dlouhodobé neaktivity informovat o možném smazání účtu z důvodu neaktivity. Je to snadný a zároveň efektivní způsob, jak systém čistit od nejspíše již nepoužívaných dat.

Uložení přihlašovacích údajů přineslo malý problém – zdá se, že `id` nelze uložit jako číslo o délce 23 číslic. Ať už se vytvoří sloupec dlouhý 32 číslic, nebo ne, SQL server v obou případech tvrdil, že je číslo příliš velké. Použití typu `BIGINT` situaci nijak nezměnilo. Kvůli tomuto zatím nevysvětlenému úkazu bylo třeba uložit číslo jak text (v SQL terminologii toto představuje typ `VARCHAR`).

## 1.2 Kniha

Tabulka knihy (v diagramu označeno jako 'book') představuje obecný vzor pro jakýkoli inzerát. Předpokládá se, že budou přidávány převážně učebnice, případně pracovní sešity, pro něž je tento termín také odpovídající.

Aby se každý hledající mohl ujistit, že našel, co hledal, vyžaduje každý inzerát název, autora i rok vydání. Samozřejmostí je pak cena, která je úmyslně omezena do 1000 Kč, neboť lze předpokládat, že žádný učební materiál jako jeden kus nebude přesahovat tuto cenu.

Už v průběhu testování se ukázalo 64 znaků pro název jako nedostatečné. Dvojnásobek se ukázal jako přijatelné množství kromě jména také pro autory, kteří nejspíše nebudou tolik ověřováni, čímž pádem bylo možné využít kratší délky a počítat s možností, že inzerenti budou muset upustit tituly autorů, pokud jich bude více než 2.

Popisek je čistě dobrovolným polem s cílem poskytnout příležitost sdělit hledajícím určité další informace, které třeba nebyly vhodné do žádné jiné položky. Příkladem může být informace o popsání knihy nebo také přímo přidaná informace, kde se pro knihu stavít, jako kupříkladu do které třídy jít za majitelem. Byť je komunikace možná přes email, někteří mohou preferovat pokus o co nepřímější a nejrychlejší způsob komunikace.

## 1.3 Žádost

Když objevíme knihu, kterou bychom si rádi pořídili, musíme si o ni zažádat. Všechny tyto žádanky se ukládají do tabulky **request**, kde uchováváme odkaz na žádajícího a knihu, o kterou žádá. Majitel knihy může přijmout pouze jednu žádost (stav přijmutí je zaznamenán ve sloupci **accepted**), může žádosti na svou knihu také rušit. Zrušit své rozhodnutí může samozřejmě také žadatel.

## 1.4 Upozornění

Upozornění (v tabulce **notification**) obvykle obsahují text k zaslání a jsou odeslána ihned. V tomto případě se jedná o princip lehce odlišný.

Název pro tento seznam se může zdát být poněkud matoucí, jedná se však o uložení textu, který uživatel, uvedený v upozornění, hledal v naději nalézt knihu. Pokud ale hledanému výrazu žádná kniha neodpovídá, je možné zažádat si o upozornění na přidání hledané knihy, čímž se myslí kniha, jejíž název obsahuje hledaný text.

## 2. Server

Všechny mechanismy a požadavky jsou zpracovávány na severu. Když uživatel chce cokoli provést, stačí zavolat správnou cestu a předat potřebná data pro dokončení akce. Server požadavek zpracuje, zapíše potřebná data do databáze a, pokud to situace vyžaduje, zašle příslušné osobě informativní email.

### 2.1 Autentifikace

Pro většinu úkonů je ale potřeba být přihlášen, na to je využíván middleware `passport`, který proces autentifikace značně zjednodušuje. Na ověření nám stačí `id` uživatele a email, protože autentifikaci k účtu jako takovému za nás řeší Google OAuth. Tento přístup je pohodlnější i pro zákazníky, jelikož nemusejí vytvářet žádný nový účet a stačí jim použít už vytvořený školní účet, který je také propojený s Googlem.

Identifikátor a email jsou ukládány v cookie prohlížeče a při přihlášení se vždy doplní o obrázek a jméno uživatele, třebaže jeho jméno není momentálně nijak využíváno. Pro pozdější komunikaci uvnitř aplikace by pak už bylo vhodné, aby uživatelé věděli, jak se navzájem oslovit. Proto ukládáme i zobrazované jméno.

### 2.2 Vstup do databáze

Na připojení k databázi je využito návrhového vzoru tovární metody (`sql.js create()`), která před vrácením metody pro zadávání příkazů databázi nejprve ověří, zda obalovou metodu k dané databázi již nevytvořila a případně ji vrátí.

Tento způsob přistupování je obecně napsaný, ale byť tato aplikace využívá pouze jednu databázi (jmenovitě `book_exchange`), člověk nikdy neví, kdy mu trocha flexibility navíc přijde vhod.

Všechny tabulky mají své fasády ve složce `services` jako soubory, které přístup do databáze dále zjednodušují svými metodami na přidání, odebrání, a zpracovávají všechny další funkce, které se s jednotlivými tabulkami pojí (např. vyhledávání knih).

### 2.3 Přístupové body

Všechny endpointy pro komunikaci se serverem mají prefix `/api`. Každá akce, která potřebuje získat nějaká data pro vykreslení stránky, přistupuje k nějaké cestě a tvoří požadavky. Tato kapitola tyto cesty, jejichž rozřazování řeší příslušné soubory pod složkou `routes` (`user.js` pro cesty `/api/user` atd.), uvede a vystvětlí práci prováděnou při jejich zavolání; všechny dále zmíněné soubory nalezneme ve složce `services`.



### 2.3.1 /user

Základní akcí pro fungování aplikace je přihlášení (/login, viz obrázek 2.1), ze kterého jsme přesměrováni passportem na přihlašovací stránku Google, která nám vrátí potřebné údaje pro přihlášení na cestu /gcallback. Zbytek práce se provádí, stejně jako u většiny ostatních cest, ve stejnojmenné metodě jako je cesta (login()) v souboru názvu příslušné tabulky, s níž pracujeme (user.js).

Pokud se jedná o registraci, tedy uživatel není nalezen v databázi, je nový účet do databáze automaticky přidán. Tím pádem není zapotřebí oddělovat registraci a přihlášení a můžeme uživatele nového i stávajícího přihlásit stejným způsobem, který zahrnuje u aktualizaci jména, či profilového obrázku, pokud by je náhodou od své poslední návštěvy změnil.

Pokud už je uživatel přihlášen (tzn. má cookie se svým emailem a id), pak pro získání všech dat spojených se svým účtem přistupuje k cestě /info, která jen vrátí záznam účtu z databáze.

Cesta /logout řeší odhlášení uživatele, kdy dochází k smazání cookie z příchozího požadavku, což efektivně odstraní z prohlížeče všechna data o uživateli.

Zavoláme-li samotnou cestu /api/user/ metodou DELETE, server náš účet z databáze smaže, čímž se smažou i všechny inzeráty a žádosti s nimi, nebo s námi, spojené. Nakonec se vyčistí i cookie z requestu, aby nedocházelo k chybě, kdy se zákazník snaží přihlásit se starými údaji, které již v databázi nejsou.

Poslední adresou je získání všech inzerovaných položek – /books. Tento bod přijímá také parametr q představující vyhledávaný výraz, podle kterého máme knihy vybrat.

Obrázek 2.1: dostupné cesty pro /user

```
/login: #GET
/gcallback: #GET
/info: #GET
/logout: #GET
/: #DELETE
/books: #GET
```

### 2.3.2 /book

Tato aplikace je o knihách, není těžké si domyslet, že se s nimi bude také nejvíce pracovat (viz obrázek 2.2). Bohužel se při práci objevily jisté problémy, kvůli kterým se musely dělat kličky.

#### Manipulace s knihou

Prvním krokem, je knihu přidat (/ #POST). Zde se neděje nic zajímavého. Když je uživatel přihlášen a předal všechna data potřebná k vytvoření knihy včetně náhledové fotky, tak knihu vytvoříme a fotku uložíme jako JPG obrázek s patřičným id. Stejný příběh je i u úpravy knihy (/ \$id #PATCH) – obdržíme všechna data, zjistíme, co je třeba pozměnit a inzerát aktualizujeme, samozřejmě úpravu může provést pouze z účtu knihu inzerující. Finálně můžeme po provedení obchodu knihu smazat (/ \$id #DELETE).

Obrázek 2.2: dostupné cesty pro /book

```
/: #POST
/$id: #PATCH
../books: #GET
/$id: #GET
/$id/edit: #GET
/$id: #DELETE
/$id/requests: #GET
```

## Data fetching

Pro získávání dat o knihách jsou k dispozici tři cesty, ale nejvíce bude v užitku adresa `/api/books` vracející všechny knihy, nebo knihy odpovídající hledanému výrazu `q`.

Druhá možnost je zeptat se na data knihy přímo, na to už je potřeba znát její id.

Konečně třetí možnost (`/$id/edit`), byť by svým názvem spíš naváděla k domněnce, že sem zasíláme upravené knihy, vrací stejná data, jako dotaz na jednu knihu. Rozdíl spočívá v kontrole, jež kontroluje právo uživatele knihu upravit.

Ačkoli se zdá, že bychom mohli kontrolu autorství provádět u klienta a není třeba na to vytvářet oddělený endpoint, v procesu vytváření frontendu se ukázala práce s uživatelem jako nemožná, přinejmenším v této části programu. Přesněji řečeno globální stav aplikace vždy při prvním zavolání vrátí prázdnou hodnotu a po získání dat o uživateli již (z nepochopitelného a nezjištěného důvodu) odmítá stránku na úpravu knížky aktualizovat, takže není způsob, jak zjistit, jestli je uživatel přihlášen.

Poslední endpoint `/$id/requests` oproti předešlým nevrací knihy, jak by se dalo očekávat, nýbrž všechny žádanky o danou knihu. Další operace s žádostmi jsou popsány níže.

### 2.3.3 `/request`

Reakce na cizí inzerát je reprezentována vytvořením nového záznamu do tabulky `request`. Všechny cesty vyobrazené v obrázku 2.3 zde dále vysvětlíme do potřebné hloubky.

Přidání neskrývá žádné změny. Pošleme číslo knihy jako parametr `book_id` a žádost propojující zákazníka s knihou se hned vytvoří spolu s emailem odeslaným majiteli knihy, aby věděl, že se někdo o jeho knihu zajímá. Více o zasílání emailů v následující kapitole 2.4.

Vlastník má navíc právo žádost přijmout (`/$id/accept`), což mu zpřístupní informace o kontaktu, přesněji jméno a email, na zákazníka, které obdrží mailem. Do té doby mají z anonymizačních důvodů přístup pouze ke svým knihám a zobrazení všech s nimi spojených žádostí, z nichž je publikováno pouze číslo žádosti a cílené knihy.

Odstranění má dvě podoby. Majitel knihy může žádost zamítnout a přímo ji smazat (`/$id #DELETE`) a žadatel zrušit (`/$id/abort`). Cílové cesty jsou rozdělené pro přehlednost kódu a snazší ověření oprávnění k provedení akce.

Poslední dvě adresy vrátí seznam žádostí, které se vztahují k Vám. Žádosti `/forMe` představují příchozí odpovědi na Vaše inzeráty. Oproti tomu `/fromMe` zobrazí Vámi odeslané žádosti na knihy uživatelů ostatních.

Obrázek 2.3: dostupné cesty pro `/requests`

```
/: #POST
/$id/accept: #PATCH
/$id: #DELETE
/$id/abort: #POST
/forMe: #GET
/fromMe: #GET
```

### 2.3.4 /notify

Endpoint `/api/notify` je výjimkou v mnoha ohledech. Nemá další podvětev a je registrován hlavním souborem serveru (`app.js`), neboť je zbytečné pro jediný přístupový bod vytvářet celý vlastní soubor, když není nijak komplexní na zpracování.

Tato adresa je volána v případech, kdy zákazník nenalezl žádnou knihu pro hledaný výraz, ale rád by obdržel upozornění o přidání knihy, která by onomu řetězci odpovídala. O to se opět postará soubor pracující s příslušnou tabulkou (`notification`) a upomínku zapíše.

Tento systém spoléhá na solidaritu uživatelů, a tedy, že nebudou vytvářet upomínky na velká množství nesmyslných názvů, které se nejspíš ve výběru nikdy neobjeví. V takové situaci by každé přidání knihy, jak bylo popsáno výše, utrhlo velké časové penále způsobené kontrolou všech upomínek.

## 2.4 Zasílání emailů

Všechny změny týkající se žádostí se vždy vztahují k dalšímu účtu, který je třeba informovat o změně. V současné chvíli pro emailování stačí jediná šablona, u které není třeba víc, než měnit titulky, podrobnější popis a odkaz tlačítka zobrazeného pod kartou knížky v jednání. Šablona je zpracovávána frameworkem `handlebars` určeným pro design emailů, které se odesílají přes plugin `nodemailer`. Podobu šablony při použití je možno vidět na obrázku 2.4.

Majitel knihy obdrží zprávu pokud:

- obdržel žádost o nabízenou knihu
- žadatel už o jeho knihu nestojí
- přijal nabídku

Žadající o knihu je informován o:

- zrušení jeho žádosti
- zmizení knihy z trhu
- přijetí zaslané nabídky
- nalezení jím hledané knihy

Pro zprávy odeslané po přijetí žádanky oběma účastníkům obchodu je navíc vygenerován výchozí text mailu na první zkontaktování protějšky strany.

**Uživatel Josef Litoš přijal Vaši žádost**  
Prosím, domluvte se s majitelem na místě předání.



Obrázek 2.4: *email obdržení po přijetí žádosti*

## 3. Webová stránka

V Předěslé části jsme se zaměřili na propojení stránky s databází za pomoci backendu, který da nás provádí všechnu práci. Díky tomu stačí na každou akci dát pokyn na patřičnou adresu a všechno se za nás zpracuje. Přesto většina problému teprve přijde, totiž práce s Reactem je, jak známo, jednoduchá na započetí, ale náročná na správné provedení.

### 3.1 Záhloví

Záhloví je přítomné ve všech cestách stránky. Máme zde k dispozici odkaz na hlavní stránku, vyhledávací pole a tlačítko uživatele, v jehož menu nalezneme možnosti



Obrázek 3.1: *záhloví stránky*

přidání knihy, zobrazení profilu a odhlášení – za předpokladu, že je uživatel přihlášen. Jinak uvidíme ikonku účtu bez profilu, která nás nabádá k přihlášení (viz obr. 3.1).

S designem aplikace nám ve všech částech pomáhá knihovna MaterialUI, která poskytuje již nastýlované komponenty, které můžeme rovnou zobrazit a nestarat se o css. Příkladem je ono vyhledávací pole, které se automaticky přizpůsobí rozměrům obrazovky a pro příliš úzké displeje zabere všechen volný prostor.

Vyhledávací pole odkazuje na seznam knih k dispozici. Výjimečným případem je vyhledávání v profilu. Potom výraz filtruje knihy, které uživatel vlastní.

### 3.2 Kniha jako stavební jednotka

Jelikož knihy zobrazujeme téměř všude, používáme na ně stejnou třídu, jen jí občas změníme parametry. Dalo by se říci, že se jedná o takový Builder, který v základu musí disponovat daty o knize spolu s náhledem.

Kniha umí přes tlačítka odkázat na svou úpravu, smazání a zažádání. První dvě akce náleží pouze vlastníkov, poslední přihlášeným uživatelům.

### 3.3 Seznam knih

Seznam knih se vyskytuje na hlavní stránce a také v profilu uživatele, kde obsahuje vlastněné inzeráty. vzhled seznamu není nijak upravovaný, knihy se zobrazí v takovém pořadí, v jakém je obdrží od backendu, což znamená, že jsou seřazené podle ceny.

Provedení hledání není šikovné, protože při každém vyhledání se nepoužije již získaný seznam všech knih, nýbrž se vyšle na server nový dotaz s hledaným výrazem a výsledek se vykreslí jako po obnovení stránky. Možné vylepšení by bylo ukládat všechny knihy do globálního stavu, a z něj potom filtrovat žádoucí položky.

Tento postup by nebyl složitý na implementaci, musel by se však kombinovat i s aktuálním přístupem, protože při otevření stránky (i obnovení) se nejprve vytváří globální stav. Toto brání programu v okamžitém vyhledání, ačkoliv by mohly knihy být načteny při prvním spuštění a později se s nimi manipulovat. Důvod byl již zmíněn v dřívější kapitole. Jedná se o vrácení prázdné hodnoty bez opětovného zavolání při obdržení nových dat.

### 3.4 Tvorba a úprava knih

Stránku na vytvoření nového inzerátu nalezneme na adrese `/commit`. Během vyplňování jednotlivých informací o knize vidíme, jak kniha bude vypadat po přidání. V tom pomáhá sdílený stav komponent potomků.

Úpravy jsou nadstavbou tvorby, kdy začínáme již s vyplněnými poli a pouze měníme hodnoty, a to včetně obrázku. Axios se o odeslání na server postará za nás, za doprovodu změny barvy tlačítka k odeslání změn, aby uživatel věděl, jestli se akce provedla úspěšně.

Každá fotka je před odesláním na server u klienta zkomprimována pomocí knihovny `client-compress` (viz Seznam Literatury), aby se šetřilo místem na disku.

### 3.5 Management žádostí

Vyžádání knihy a dále stav jednání můžeme sledovat v tabulkách na stránce našeho profilu (`/user`). Tabulky také obsahují tlačítka s akcemi, přejetím myši získáme podrobnější informace o akci (viz obr. 3.2)

V žádostech o naše produkty můžeme logicky přijmout jen jednu. Proto pokud již nějaká žádost pro danou knihu je přijata budou ostatní skryté. Až kdybychom již přijatou žádost zrušili, třeba z důvodů příliš dlouhé domluvy, se nám zobrazí ostatní od stejné knihy.

### 3.6 Překážky během vývoje

První nárazkou na nepříjemnosti je nejednotnost návodů a pomocných fór na internetu. Základní balíčky Reactu (`react-router-dom`, `react-redux`...) prošli nedávno zásadnějšími změnami, které nejsou dobře dokumentované a jejich chování může překvapit. Jiné funkce naproti tomu nejsou konzistentní (zmiňované updatování globálního stavu). K tomu ještě chybí znalost základů, kterých jsme měli nabýt ve třetím ročníku, a výsledkem je program, který nemá žádné optimalizace.

Příchozí objednávky			
Volby	Stav	Datum zažádání	Název knihy
	Čeká na přijetí	14. 2. 2022 22:01:42	Matematika pro gymnázia - Analytická geometrie
	Předáváte	26. 2. 2022 21:57:27	Matematika pro gymnázia - Goniometrie

Moje objednávky			
Zrušit	Stav	Datum zažádání	Název knihy
	Přijato	12. 2. 2022 21:44:10	Matematika pro gymnázia - Planimetrie
	Přijato	26. 2. 2022 21:52:04	Matematika pro gymnázia - Stereometrie

Obrázek 3.2: *vrchní část profilové stránky*

## Závěr

Program funguje, z pohledu designu je svou jednoduchostí i poměrně pěkný. Vše, co jej dělá užitečným, už pracuje tak, jak má, a to stabilně – s případnými záchytnými body.

Ačkoliv provozuschopnost aplikace dosáhla navrženého minima, vyskytuje se v kódu i nadále mnoho věcí, které by stálo za to přidat, či v několika případech také zlepšit (filtrování knih na straně klienta). Příkladem toho je navržení pro malé rozměry – v aktuální podobě vždy pošleme seznam všech knih v databázi, namísto posílání po částech na vyžádání.

K větším negativním vlastnostem patří scházějící handlování o ceny, které lze napodobit úpravou ceny uvedené na inzerátu, což docílí stejného efektu, ale ve většině případů pravděpodobně nebude příliš využíváno.

Podobný dopad má skrytí kontaktních údajů do doby přijetí. Uživatel nemá možnost komunikovat s více lidmi současně a dát knihu rychlejšímu. Na druhou stranu toto osobně považuji za pozitivní. Vyhýbáme se tím situacím, kdy by uživatelé zbytečně marnili čas s jedním inzerentem, zatímco takto mohou požádat u více lidí, a obchod provádět s tím, který jako první odpoví jim. Tímto se přenáší soutěživost mezi prodávající namísto kupců, což odpovídá mnohdy i skutečnosti.

Bonus v podobě designu pro mobilní zařízení považuji za dosažený – Všechny ukázky z programu byly v rozměrech telefonů a nevidím v nich chybu. Oproti tomu pohled na počítači by mohl zužitkovat menšího zlepšení, avšak fotky beztak budou nahrávány převážně z elektroniky vybavené fotoaparátem.

Výhodnější balíčky by určitě stály za doprogramování, ale další tabulka spolu s novým odkazem pro zobrazení obsahu by měla cenu jen, pokud by program byl využíván.

Administrátor by do databází neměl mít potřebu zasahovat, a pokud ano, jednalo by se pravděpodobně o stav, kdy by aplikace stejně nemohla běžet a byl by zapotřebí přímý přístup, což prakticky eliminuje pointu správy z prostředí webové stránky.

Kdyby byla šance, že tato aplikace by běžela pod určitou doménou, stálo by za to do budoucna implementovat i chat do stránky, aby komunikace probíhala pohodlněji. To by bohužel znamenalo buď větší spam emailů po každé zprávě protější strany, nebo nutnost stránku neustále kontrolovat, takže výhoda této funkce je pochybná.

# Seznam použité literatury

*How to Auto-resize an Image to Fit into a DIV Container using CSS.* Dostupné z <https://www.tutorialrepublic.com/faq/how-to-auto-resize-an-image-to-fit-into-a-div-container-using-css.php>. [cit 2022-10-2].

AISELVAM, T. *Node Multer unexpected field.* Dostupné z <https://stackoverflow.com/a/31532067/12174842>. [cit 2022-15-1].

*Sending Email via Gmail SMTP Server in NodeJS.* Artisans Web. Dostupné z <https://artisansweb.net/sending-email-via-gmail-smtp-server-in-nodejs/>. [cit 2022-18-2].

BAILEY, D. *URI/URL Encoding Strings In JavaScript/Node.js.* Dostupné z <https://derickbailey.com/2016/05/02/uriurl-encoding-strings-in-node-js/>. [cit 2022-22-1].

BONE, A. *How to Use Aliases with SQL JOINS.* Dostupné z <https://learnsql.com/blog/how-to-use-aliases-with-sql-join/>. [cit 2022-14-2].

COPE, S. *:not / CSS-Tricks.* Dostupné z <https://css-tricks.com/almanac/selectors/n/not/>. [cit 2022-25-1].

DAMIEN. *LIKE vs CONTAINS on SQL Server.* Dostupné z <https://stackoverflow.com/a/7510685/12174842>. [cit 2022-15-1].

DAVEJM. *client-compress.* Dostupné z <https://github.com/davejm/client-compress>. [cit 2022-2-2].

*Using the Public Folder.* Facebook, Inc. Dostupné z <https://create-react-app.dev/docs/using-the-public-folder/>. [cit 2022-2-2].

FEDYASHEV, N. *axios post request to send form data.* Dostupné z <https://stackoverflow.com/a/47630754/12174842>. [cit 2022-3-2].

FRANKLIN, J. *Reading and updating query params with URLSearchParams.* Dostupné z <https://www.jackfranklin.co.uk/blog/url-search-params/>. [cit 2022-7-2].

HERRERA, E. *React Router DOM: How to handle routing in web apps.* LogRocket. Dostupné z <https://blog.logrocket.com/react-router-dom-tutorial-examples/>. [cit 2022-28-1].

HUR, C. *How to Access URL Parameters Using React Router 6.* Dostupné z <https://chrisvhur.medium.com/how-to-access-url-parameters-using-react-router-6-a4cf6bdad4dd>. [cit 2022-9-2].

- IBRAHIM. *Custom Upload Button, Image Preview and Image Upload with React Hooks*. Dostupné z <https://medium.com/@ibamibrhm/custom-upload-button-image-preview-and-image-upload-with-react-hooks-a7977618ee8c>. [cit 2022-30-1].
- KUNUKN. *input's event.target is null within this,setState [React.js]*. Dostupné z <https://stackoverflow.com/a/47443091/12174842>. [cit 2022-2-1].
- MUI: *The React component list*. Material-UI SAS. Dostupné z <https://mui.com/components>. [cit 2022-15-1].
- MCGINNITS, T. *A Guide to Query Strings with React Router*. Dostupné z <https://ui.dev/react-router-query-strings/>. [cit 2022-7-2].
- Forms*. Meta Platforms, Inc. Dostupné z <https://reactjs.org/docs/forms.html>. [cit 2022-28-1].
- Child combinator*. Mozilla. Dostupné z [https://developer.mozilla.org/en-US/docs/Web/CSS/Child\\_combinator](https://developer.mozilla.org/en-US/docs/Web/CSS/Child_combinator). [cit 2022-5-2].
- Creating a Material-UI form*. One Step! Code. Dostupné z <https://onestepcode.com/creating-a-material-ui-form/>. [cit 2022-28-1].
- Serving static files in Express*. OpenJS Foundation. Dostupné z <https://expressjs.com/en/starter/static-files.html>. [cit 2022-10-2].
- JavaScript String search()*. Refsnes Data. Dostupné z [https://www.w3schools.com/jsref/jsref\\_search.asp](https://www.w3schools.com/jsref/jsref_search.asp). [cit 2021-30-12].
- SKINNER, F. *Programmatically navigate using React router*. Dostupné z <https://stackoverflow.com/a/31079244/12174842>. [cit 2022-16-2].
- STORM, A. *How to Link in React and the Material UI Framework*. Dostupné z <https://alanstorm.com/how-to-link-in-react-and-the-material-ui-framework/>. [cit 2022-20-1].
- THOMAS, J. *How to change the state of a child component from its parent in React*. Dostupné z <https://www.freecodecamp.org/news/react-changing-state-of-child-component-from-parent-8ab547436271/>. [cit 2022-5-2].
- Library logo*. Tradefair Hotels. Dostupné z <http://www.tradefair-hotels.com/home/attachment/library-logo/>. [cit 2021-27-12].



# Seznam obrázků

1.1	<i>Diagram používané databáze</i>	5
2.1	<i>dostupné cesty pro /user</i>	8
2.2	<i>dostupné cesty pro /book</i>	8
2.3	<i>dostupné cesty pro /requests</i>	9
2.4	<i>email obdrženy po přijetí žádosti</i>	10
3.1	<i>záhlaví stránky</i>	11
3.2	<i>vrchní část profilové stránky</i>	12