

АКАДЕМИЈА ТЕХНИЧКО-УМЕТНИЧКИХ СТРУКОВНИХ
СТУДИЈА БЕОГРАД
ОДСЕК ВИСОКА ШКОЛА ЗА ИНФОРМАЦИОНЕ И
КОМУНИКАЦИОНЕ ТЕХНОЛОГИЈЕ

Веб апликација за коришћење и управљање књижаром

ЗАВРШНИ РАД

Ментор:
мр Миланко Краговић

Кандидат:
Алекса Берисавац 32-18

Београд, 2024.

[Type here]

АКАДЕМИЈА ТЕХНИЧКО-УМЕТНИЧКИХ СТРУКОВНИХ
СТУДИЈА БЕОГРАД
ОДСЕК ВИСОКА ШКОЛА ЗА ИНФОРМАЦИОНЕ И
КОМУНИКАЦИОНЕ ТЕХНОЛОГИЈЕ

Интернет технологије

Предмет: Web programiranje ASP

Тема: Веб апликација за коришћење и управљање књижаром

Ментор:
мр Миланко Краговић

Кандидат:
Алекса Берисавац 32-18

Београд, 2024.

[Type here]

Садржај:

1. Uvod.....	4
1.1 Korišćeni programski jezici/tehnologije	4
1.2 Opis funkcionalnosti	4
1.3. Napomene za pregledanje	Error! Bookmark not defined.
2. Slike stranica i opisi funkcionalnosti	6
2.1 Home	6
2.2 Books.....	7
2.3. Cart.....	7
2.4. Checkout	9
2.5. 404 – Not Found.....	10
2.6. Modeli	10
2.7. Struktura tsBusinessLayer-a.....	12

1. УВОД

1.1 Коришћени програмски језици и технологије

За потребе креирања апликације коришћене су следеће технологије и алати:

- HTML, CSS
- JavaScript (TypeScript)
- Angular 16
- Router module
- Material Design
- VS Code
- Visual Studio
- .NET Framework
- Entity Framework
- AutoMapper
- Fluent Validation
- Newtonsoft

1.2 Опис функционалности:

Домен који апликација покрива је домен једне продавнице књига. Корисници могу да разгледају производе и да их додају у корпу. Потом могу да изврше куповину, ако су улоговани. Са обзиром на то да је у фокусу пројекта коришћење и управљање књижаром са административне стране, овим се ток коришћења за корисника који није администратор завршава. Корисник улогован као администратор може руководити целокупним садржајем као и вршити преглед наруџбина свих корисника.

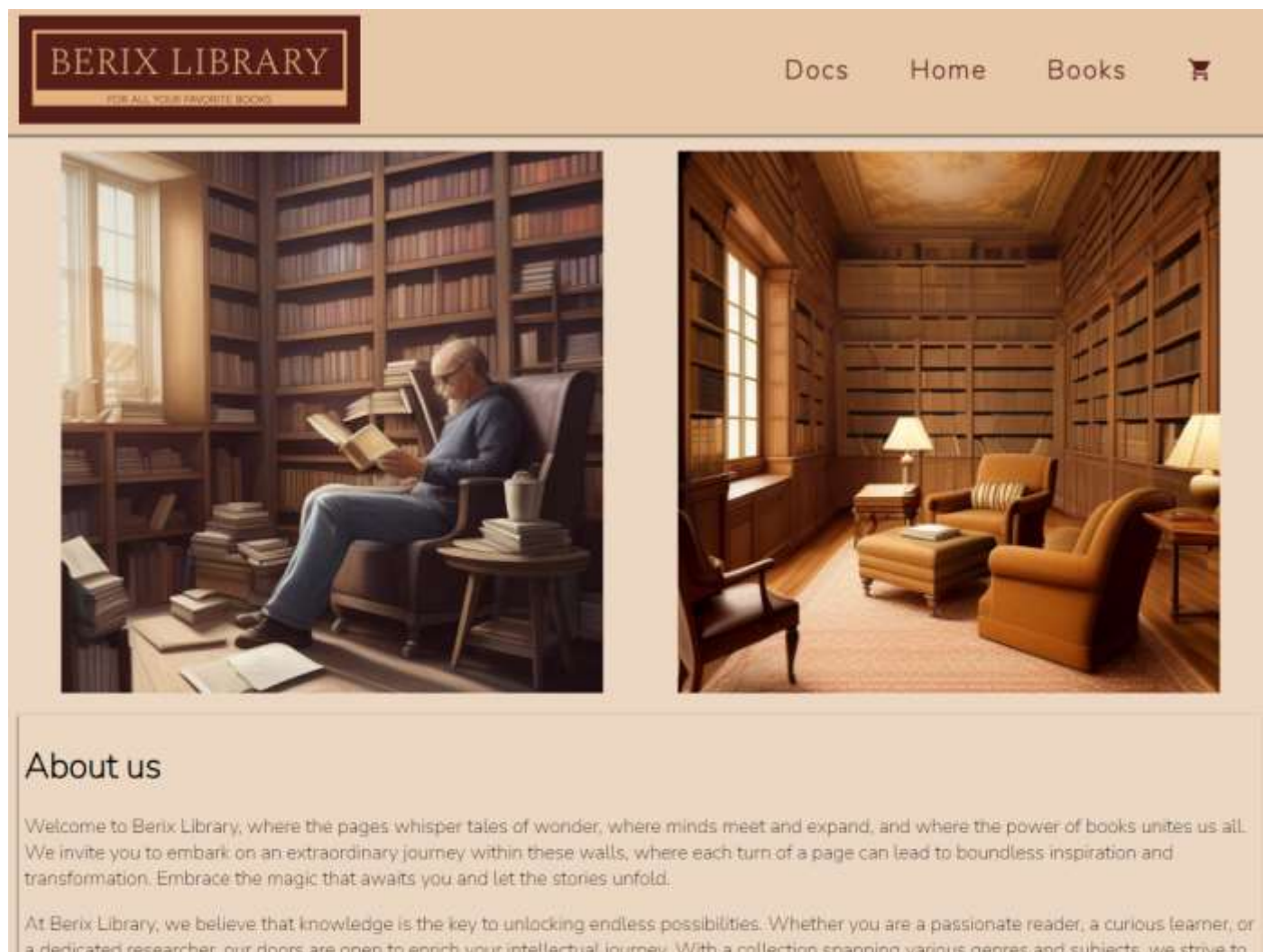
[Type here]

Функционалности сајта се могу сврстати у следеће категорије:

- **Респонзиван дизајн.**
- **Почетна страница**, динамички учитава текстуалне податке за приказ.
- **Рутирање.**
- **Галерија књига** са динамичким исписом појединих књига и њених детаља.
- **Страница за приказ детаља о књизи.**
- **Страница за приказ детаља о аутору.**
- **Администрирање над садржајем сајта**, све сем постављања слике за књигу.
Имплементирана је слика која се приказује за овај случај.
- **Корпа** која смешта податке у сервис, са укупном сумом и опцијама за брисање и измену количине.
- **Форма за куповину**
- **55** случајева коришћења.
- **17** табела.
- **Логовање** свих случајева коришћења са све улазним подацима, ако постоје.

2. Slike stranica i opisi funkcionalnosti

2.1 Home

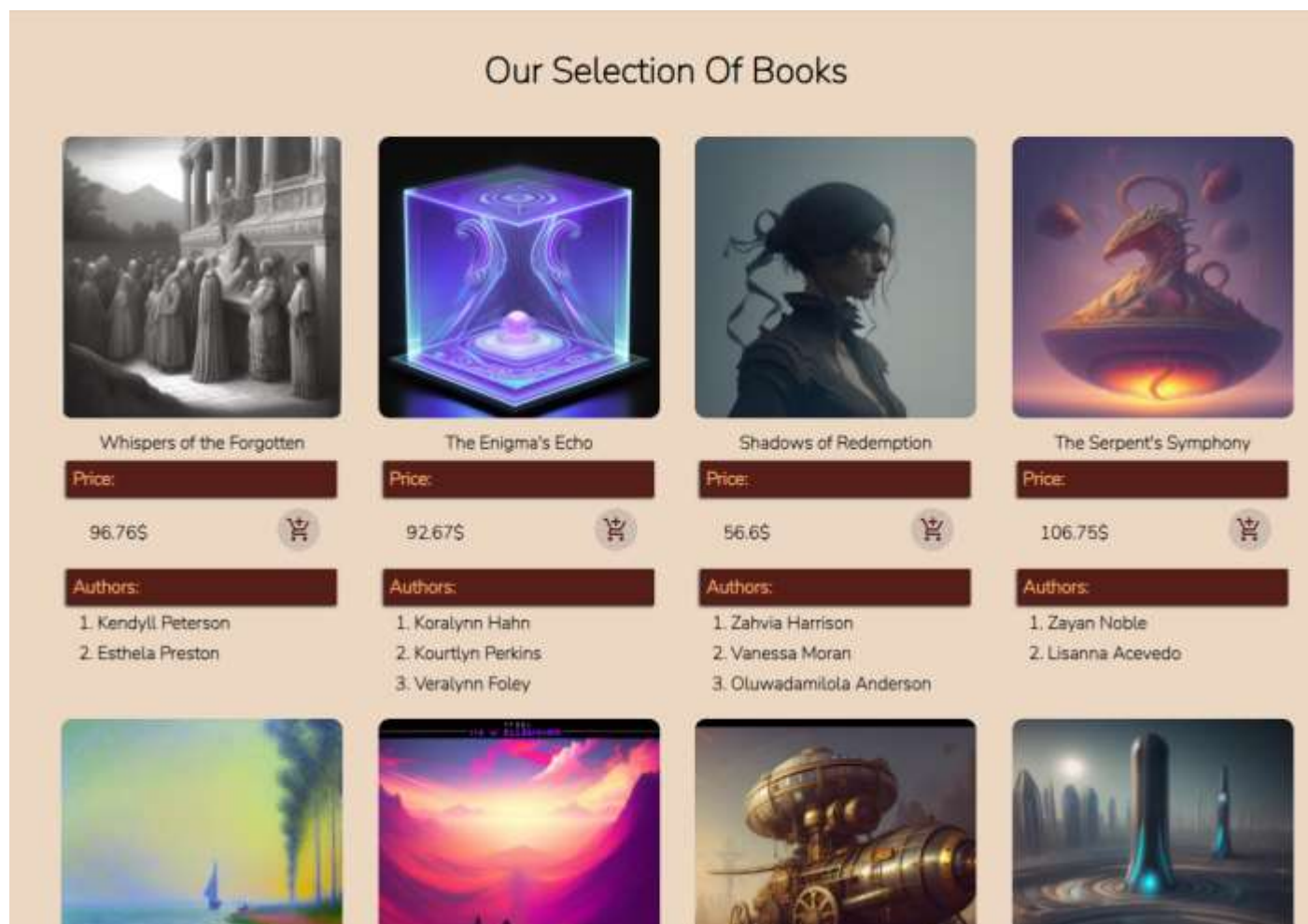


Na ovoj stranici se nalazi i navigacija, koja se zbog jako malog broja elemenata ne učitava ni iz kakvog json objekta. Ona koristi rutiranje, i responzivna je.

Tekst koji se ispisuje na početnoj stranici se učitava dinamički iz json-a, u vidu niza paragrafa.

[Type here]

2.2 Books



Na ovoj stranici se mogu videti sve knjige. Knjige se ucitavaju preko servisa koji komunicira sa tsBusinessLayer-om, preko odgovarajucih modela, koji uzimaju podatke iz json-a. Potom se knjige ispisuju u komponenti dinamicki. Tu se takodje nalazi dugme za dodavanje u korpu. Nakon klika na ovo dugme stvori se jos jedno dugme koje sluzi za uklanjanje jednog tog item-a iz korpe. Ono sto je ostalo neimplementirano zbog nedostatka vremena jesu parcijalne stranice za autora I knjigu, a sto se tice dizajna da se iskoriste slideUp I slideDown za autore kako bi bilo preglednije. Svi neophodni podaci za ove funkcionalnosti su naravno tu, knjige imaju svoje zanrove, opise I jos dosta toga.

2.3. Cart

[Type here]

Your Shopping Cart:

There are currently no items in your cart. [You can go to the books page and see what you like.](#)

BERIX LIBRARY


FOR ALL YOUR FAVORITE BOOKS

[Docs](#)

[Home](#)

[Books](#)

 | 6

Your Shopping Cart: 259\$  [checkout](#)

Art	Title	Price	Total Price	Quantity	Remove	Add
	Whispers of the Forgotten	96.76\$	96.76\$	#1		
	Whispers in the Mist	44.19\$	132.57\$	#3		
	The Labyrinth's Secret	16.09\$	16.09\$	#1		
						

Na ovoj stranici se vidi sadržaj korpe korisnika (ako nema proizvoda u korpi onda se izdaje takva poruka).

Ukupna vrednost se izračunava i prikazuje.

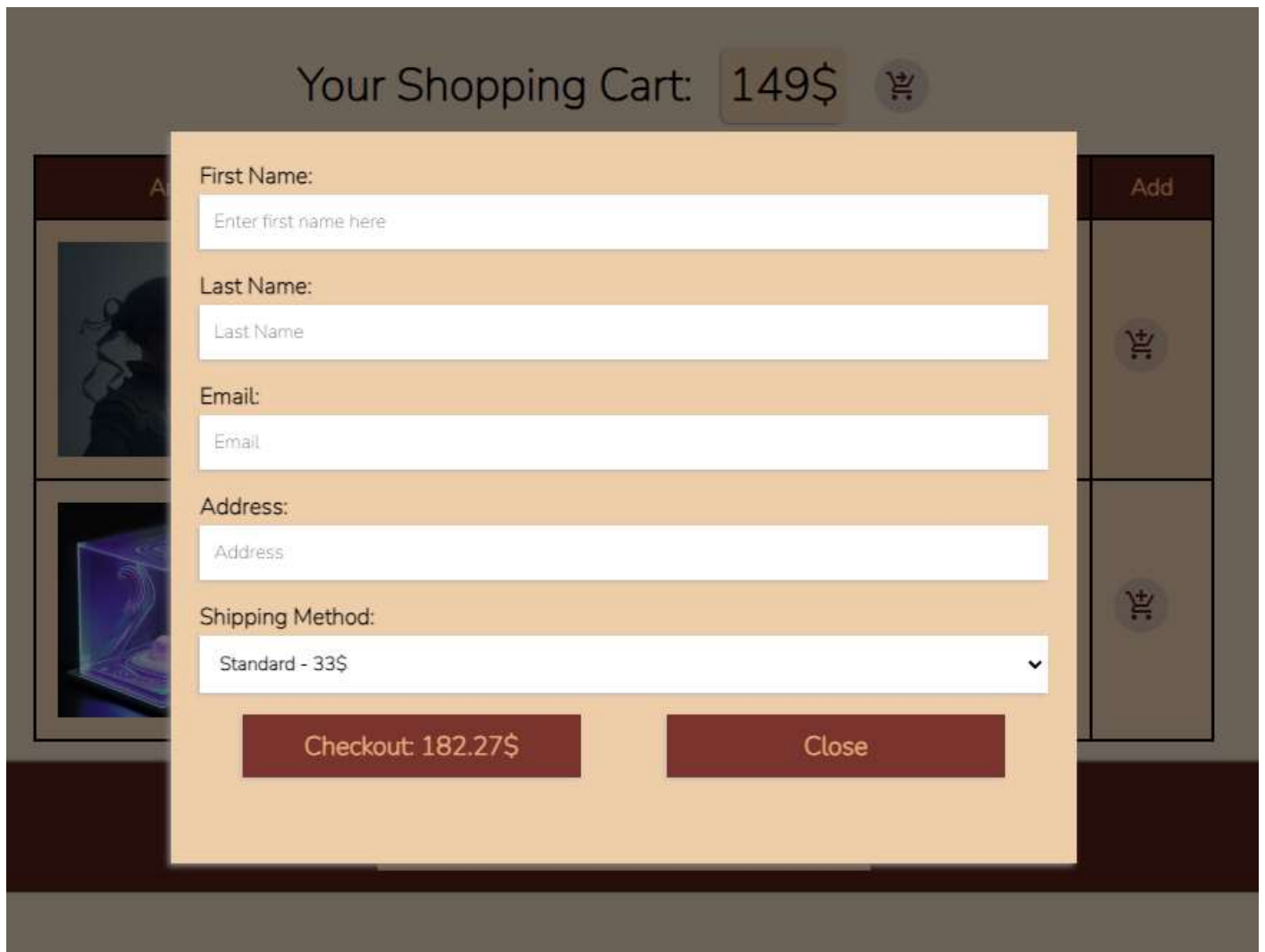
Korpa je napravljena kao servis koji se koristi na više mesta, od kojih su neki: navigacija (gde se pored ikonice korpe vidi broj itema trenutno u njoj), sama komponenta korpa, checkout, prikaz knjiga u books rutini..

Ovde je takođe moguće uklanjati i dodavati proizvode.

[Type here]

Klikom na dugme za checkout otvara se modal sa formom za unos podataka za dostavu. Nije mi radio bas kako sam hteo Material Modal sa svojim inputima pa sam napravio svoje. Ukoliko se ta forma popuni ispravno korpa se čisti i korisnik se obavestava o pozitivnom ishodu narudžbine. U suportnom dobija poruku da nije lepo popunio podatke. Zbog manjka vremena nisam previse vremena posvetio validaciji.

2.4. Checkout



The image shows a web application interface with a shopping cart and a checkout modal. At the top, the text "Your Shopping Cart: 149\$" is displayed next to a shopping cart icon. Below this, there is a table with two columns: "Add" and "Remove". The table contains two rows, each with a product image and a "Remove" icon. A modal window is open in the center, titled "Checkout: 182.27\$". The modal contains several input fields: "First Name:" (placeholder: "Enter first name here"), "Last Name:" (placeholder: "Last Name"), "Email:" (placeholder: "Email"), "Address:" (placeholder: "Address"), and "Shipping Method:" (dropdown menu with "Standard - 33\$" selected). At the bottom of the modal, there are two buttons: "Checkout: 182.27\$" and "Close".

U ovoj komponenti se unose podaci o korisniku. Ideja je bila da se ovi podaci cuvaju, ali posto na frontu ne moze da se edituje json fajl zbog security-ja to nije odradjeno. Svakako je predvidjeno kao da jeste, I u svrhu simulacije I završavanja ovog flow-a ne moze se naruciti ako se ne unesu bar neki podaci.

Na dugmetu za checkout pise ukupna cena sa selektovanom postarinom.

[Type here]

Na klik na checkout, posle uspesne validacije, formira se promenljiva tipa FormData, I appenduju se svi potrebni podaci u nju. Sa njom se nista ne radi iz gore navedenih razloga, ali tu je u svrhu demonstriranja formata promenljive koja bi se slala na server.

Jos jednom ponavljam, celokupna struktura podataka sto se tice ordera I orderInvoice-a se moze videti u dto sloju unutar tsBusinessLayer foldera. Jedino sto nije implementirano jesu modeli za Usera I za Order, iz gore navedenih razloga.

2.5. 404 – Not Found



Pri pokusaju pristupanja stranici koja nije definisana u ruter modelu prikazuje se 404 komponenta.

2.6. Modeli

Primer modela za knjigu:

[Type here]

```

declare var require: any;

import { BookPriceDTO } from "../dtos/BookPriceDTO";
import { IEntityGet } from "../interfaces/common/IEntityGet";
import { IEntityGetAll } from "../interfaces/common/IEntityGetAll";
import { TBookJSON } from "../types/JSON/TBookJson";
import { TBookPriceJson } from "../types/JSON/TBookPriceJson";

export class BookPriceModel implements IEntityGetAll, IEntityGet{
    private bookPrices: Array<BookPriceDTO> = [];

    constructor(){
        let bookPrices: Array<TBookPriceJson> = require("src/assets/data/bookPrice.json")
        this.bookPrices = this.convertAuthorFromJsonToDTO(bookPrices);
    }

    public getAll<BookPriceDTO>(): Array<BookPriceDTO> {
        return this.bookPrices as Array<BookPriceDTO>;
    }

    public get<BookPriceDTO>(id: number): BookPriceDTO {BookPriceDTO
        return this.bookPrices.filter(x=>x.id==id)[0] as BookPriceDTO;
    }

    public getBookPricesByBook(book: TBookJSON): Array<BookPriceDTO>{
        let bookPriceIdsArray = book.BookPrices;

        let bookPrices: Array<BookPriceDTO> = [];
        for(let bookPriceId of bookPriceIdsArray){
            bookPrices.push(this.get(bookPriceId))
        }

        return bookPrices.sort((x,y)=>{return x.CreatedAt>y.CreatedAt? -1: 1});
    }

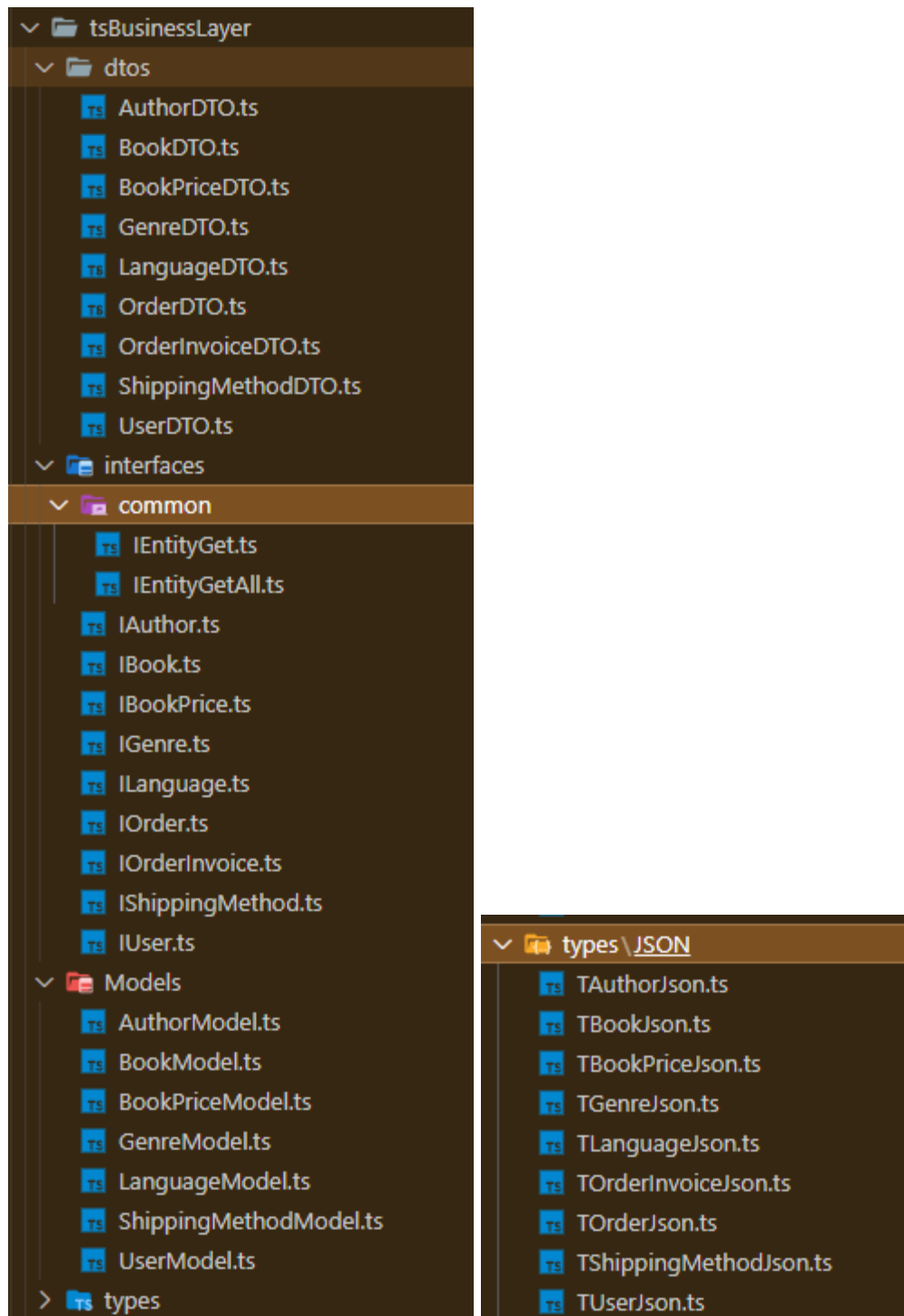
    private convertAuthorFromJsonToDTO(bookPrices: Array<TBookPriceJson>): Array<BookPriceDTO>{
        let bookPricesDTO: Array<BookPriceDTO> = [];
        for(let bookPrice of bookPrices){
            bookPricesDTO.push({
                "id": bookPrice.id,
                "Price": bookPrice.Price,
                "CreatedAt": new Date(`${bookPrice.CreatedAt}`)
            })
        }

        return bookPricesDTO;
    }
}

```

[Type here]

2.7. Struktura tsBusinessLayer-a



[Type here]