

Odsek Visoka ICT škola,
Akademija tehničko-umetničkih strukovnih studija
Beograd

Dokumentacija

Bookstore

Smer: Internet tehnologije
Modul: Web programiranje
Predmet: Web programiranje PHP 2

Dorđe Solomun 16/19

Contents

1. Uvod	5
1.1 Korišćene tehnologije	5
1.2 Opis funkcionalnosti	5
1.3 Template	6
2. Organizacija	6
2.1 Dizajn baze podataka	6
2.2 Struktura projekta	7
2.3 Slike stranica i opis funkcionalnosti	9
2.3.1 Home za korisnike i goste	9
2.3.2 Shop	10
2.3.3 Stranica za prikaz jedne knjige	13
2.3.4 Contact	14
2.3.5 Register	15
2.3.6 Login	16
2.3.7 Author	17
2.3.8 Cart	17
2.3.9 Admin home	19
2.3.10 Admin products	19
2.3.11 Add new book	20
2.3.12 Edit book	20
2.3.13 Admin genres	21
2.3.14 Add new genre	21
2.3.15 Edit genre	22
2.3.16 Admin users	22
3. Kodovi	23
3.1 Kontroleri	23
3.1.1 AdminController	23
3.1.2 AuthController	24
3.1.3 BaseController	25
3.1.4 BookController	26
3.1.5 CartController	29
3.1.6 ContactController	31
3.1.7 Controller (Laravelov kontroler nije menjan)	32
3.1.8 GenreController	32
3.1.9 HomeController	33

3.2. Middleware	34
3.2.1 IsAdmin	34
3.2.2 IsGuest	34
3.2.3 IsLoggedIn	35
3.3 Requests.....	35
3.3.1 BookStoreRequest.....	35
3.3.2 BookUpdateRequest	36
3.3.3 ContactRequest.....	37
3.3.4 GenreStoreRequest.....	38
3.3.5 GenreUpdateRequest	38
3.3.6 LoginRequest.....	39
3.3.7 RegisterRequest	40
3.4 Models	40
3.4.1 Activity	40
3.4.2 Author	41
3.4.3 Book	41
3.4.4 Contact	41
3.4.5 Genre.....	41
3.4.6 Role	42
3.4.7 User	42
3.5 Migrations.....	42
3.5.1 Roles table.....	42
3.5.2 Users table	43
3.5.3 Genres table.....	44
3.5.4 Authors table.....	44
3.5.5 Books table.....	45
3.5.6 Author_book table	45
3.5.7 Contacts table	46
3.5.8 Activities table.....	46
3.6. Seeders.....	47
3.6.1 BookSeeder	47
3.6.2 DatabaseSeeder	63
3.6.3 GenreSeeder	63
3.6.4 RoleSeeder	64
3.6.5 UserSeeder.....	64
3.7 Views	65

3.7.1 Fixed	65
3.7.2 Layouts	68
3.7.3 Pages	68
3.8 Routes	88
3.8.1 web.php	88
3.9 CSS.....	89
3.9.1 style.css	89
3.10 JavaScript	93
3.10.1 main.js.....	93

1. Uvod

1.1 Korišćene tehnologije

- HTML 5
- CSS 3
- JavaScript
- jQuery
- Toastr
- Bootstrap v4.5
- Font Awesome
- Heroku za online bazu
- MySQL Workbench za rad sa bazom
- PHP
- Laravel 8 PHP Framework

Kucano u PhpStorm.

Baza postavljena online na Heroku sajt.

1.2 Opis funkcionalnosti

- Sajt je responsive i prilagođen prikazu na svim ekranima.
- Navigacioni meni je responsive, na manjim rezolucijama se prikazuje hamburger ikonica na koju se klikne da bi se meni proširio. Realizovano bootstrapom.
- Linkovi u navigacionom meniju se menjaju u zavisnosti od toga da li je neko ulogovan i ako je ulogovan da li je običan korisnik ili admin.
- Prodavnica sa proizvodima koji se dinamički ispisuju iz baze, dohvaćeni AJAX-om i ispisani JavaScript-om.
- Elementi za filtriranje po žanru ispisani dinamički podacima iz baze.
- U donjem desnu uglu sajta se pojavljuje dugme za skrol na vrh kada se skroluje 200 pixela na dole.
- Paginacija proizvoda, usklađena sa filtriranjem po žanru, sortiranjem i pretragom po imenu knjige ili imenu autora knjige.
- Slanje poruke na kontakt strani, validacija sa serverske i klijentske strane.
- Registracija korisnika, validacija forme i na klijentskoj i na serverskoj strani.
- Login korisnika, validacija forme i na klijentskoj i na serverskoj strani. Nakon login-a obični korisnici se šalju na početnu stranu sajta za korisnike, a admin se šalje na admin panel.
- Ulogovani korisnik može da dodaje proizvode u korpu. Admin ne može. Realizovano pomoću AJAX-a i Toastr ispisuje adekvatnu poruku o uspehu ili neuspehu.
- Korisnik može da pregleda svoju korpu koja se čuva u sesiji i odatle dinamički ispisuje. Može da menja količinu proizvoda u korpi, da ih briše iz korpe i da poruči. JavaScript i AJAX realizuju ove promene količine i dinamički osvežavaju prikaz ukupne cene i broja proizvoda u korpi. Samo ulogovani korisnici mogu pristupiti stranici za pregled korpe.
- Zapisivanje ključnih događaja kao što su registracija novog korisnika, login, logout, dodavanje u korpu, poručivanje, dodavanje novih proizvoda, menjanje proizvoda, dodavanje žanrova, menjanje žanrova, brisanje korisnika u tabelu u bazi.
- Admin panel sa prikazom svih ključnih aktivnosti od strane korisnika, mogućnost sortiranja po datumu. Prikaz svih proizvoda, žanrova i mogućnost dodavanja novih, menjanja ili brisanja postojećih. Prikaz svih korisnika i mogućnost brisanja korisnika. Prikaz svih poruka koje su poslate sa kontakt strane.
- Admin panelu može da pristupi samo admin, ako običan korisnik pokuša biće redirektovan. Admin može da pristupi korisničkim stranicama, ali ne može da dodaje u korpu i da pristupi stranici za pregled korpe.

- Admin stranice i korisničke stranice koriste drugačije layout fajlove.

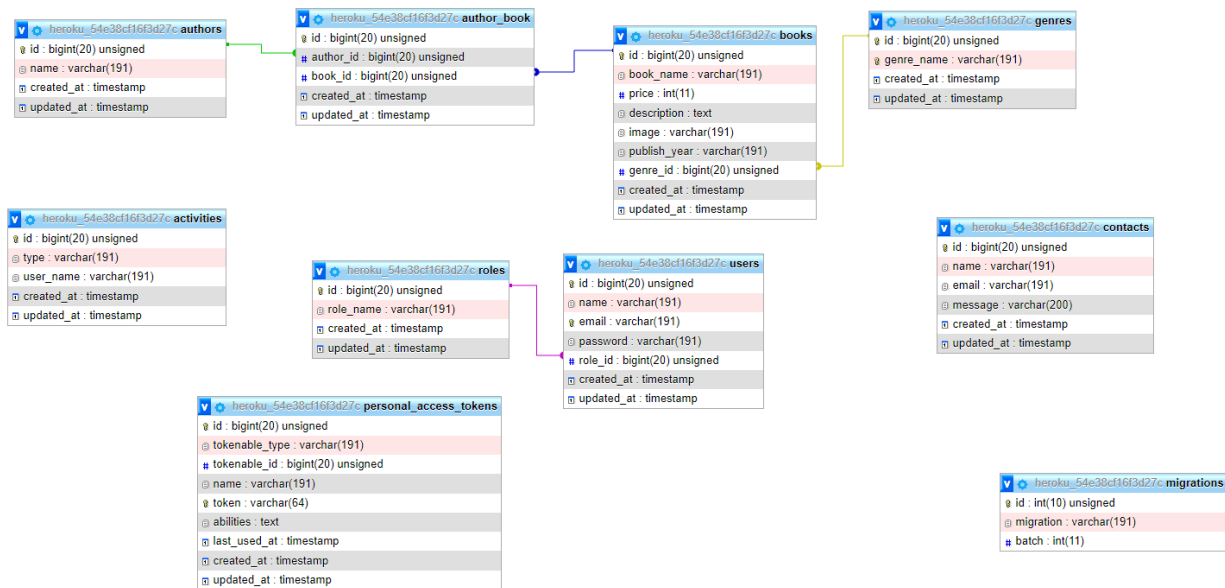
Sajt treba da realizuje online prodaju knjiga, uz admin panel da admin može da upravlja sadržajem i korisnicima sajta. Korisničke stranice i admin stranice koriste drugačiji layout. Korisnici ne mogu da pristupe stranicama za admina, a admin može da pristupi admin panelu i svim korisničkim stranicama samo nema pristup njihovim funkcionalnostima kao što je dodavanje u korpu i pregled korpe.

1.3 Template

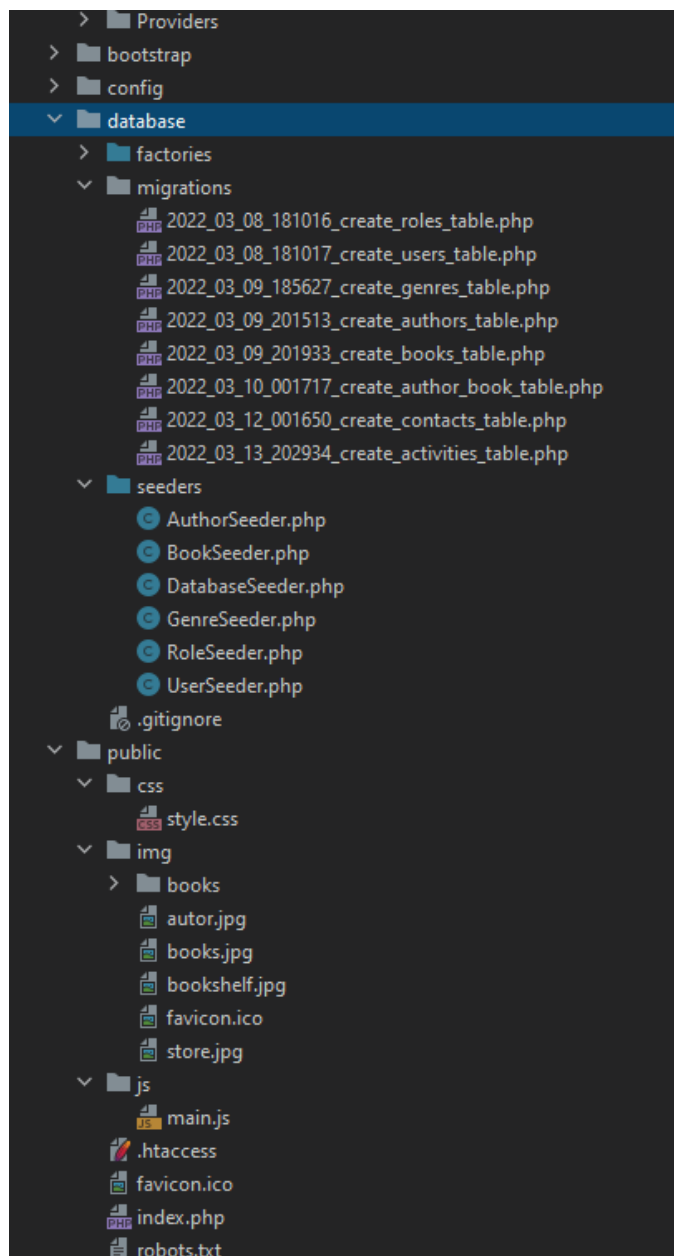
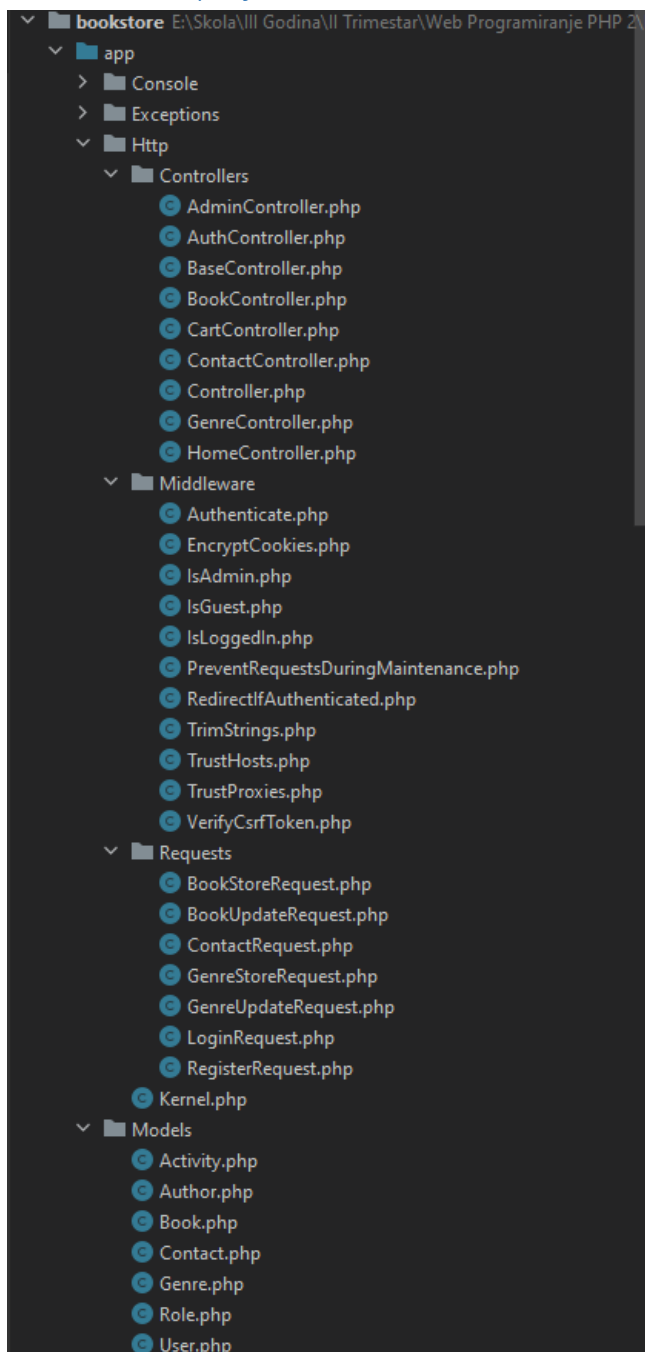
Nije korišćen gotov template, kucan je html,css,js i php kod da bi se realizovao izgled.

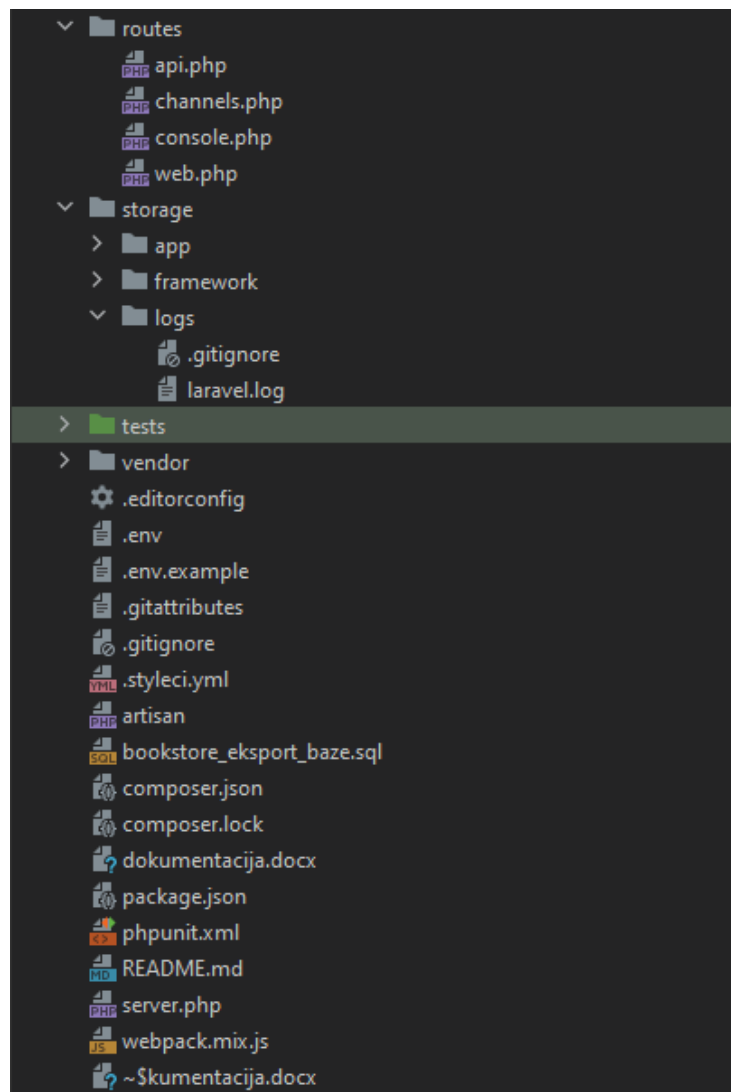
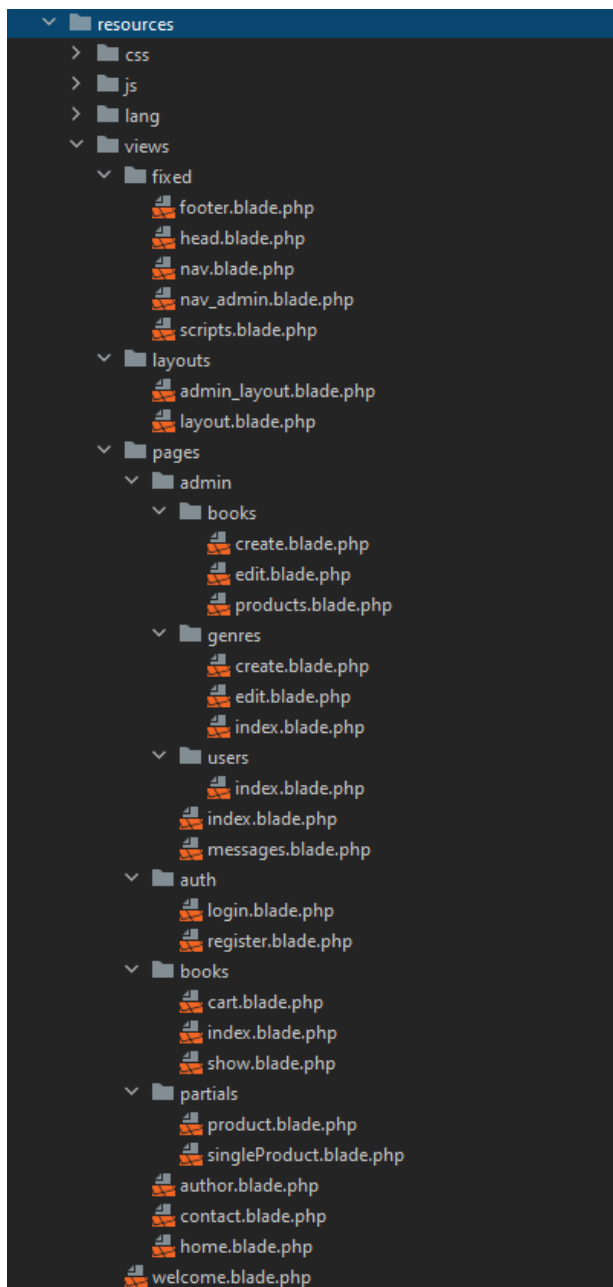
2. Organizacija

2.1 Dizajn baze podataka



2.2 Struktura projekta





2.3 Slike stranica i opis funkcionalnosti

2.3.1 Home za korisnike i goste

Navigacioni meni se menja u zavisnosti od toga da li je ulogovan korisnik ili admin. Kad je neko ulogovan pojavljuje se logout dugme pored kojeg piše ime i prezime korisnika. Navigacioni link je obojen ako je ta stranica trenutno aktivna. Na stranici se nalazi statički sadržaj i blok u kome se prikazuju četiri najskorije dodata proizvoda. Ovo je prikaz kada niko nije ulogovan.



PC Center



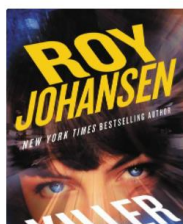
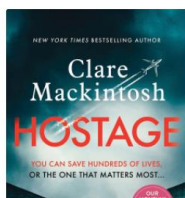
PC Center

If you like to read this is the place for you. We have something for everyone, and we are always expanding our offer.

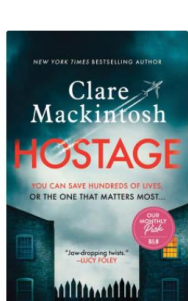
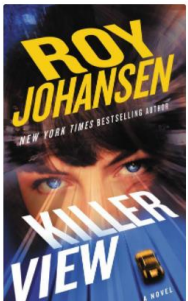
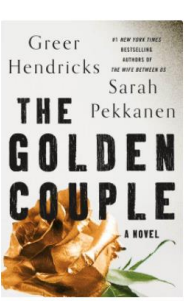

Take a look at our book collection: [shop](#)



Newest releases



Newest releases

 <p>Hostage</p> <p>Genre: Thriller</p> <p>Author: Clare Mackintosh</p> <p>Price: \$18</p> <p>Add to cart</p>	 <p>Killer View</p> <p>Genre: Thriller</p> <p>Author: Roy Johansen</p> <p>Price: \$8</p> <p>Add to cart</p>	 <p>The Golden Couple</p> <p>Genre: Thriller</p> <p>Author: Greer Hendricks, Sarah Pekkanen</p> <p>Price: \$15</p> <p>Add to cart</p>	 <p>The Overnight Guest</p> <p>Genre: Thriller</p> <p>Author: Heather Gudenkauf</p> <p>Price: \$18</p> <p>Add to cart</p>
--	---	---	---

<p>Follow us</p> <p>Facebook Instagram YouTube</p>	<p>Working hours</p> <p>Workdays: 08-20h Weekends: 09-18h Knez Mihailova 1, Belgrade</p>	<p>Links</p> <p>Documentation Author ©Djordje Solomun 16/19</p>
---	---	--

Navigacioni meni kada se uloguje korisnik.

<p>Bookstore Home Shop Contact Cart</p>	<p>Petar Petrovic Logout</p>
---	------------------------------

Navigacioni meni kada se uloguje admin. Ovo je korisničkoj home stranici na stranicama za admina navigacija je drugačija jer te stranice koriste drugi layout.

<p>Bookstore Home Shop Contact Admin panel</p>	<p>Marko Markovic Logout</p>
--	------------------------------

2.3.2 Shop

Prikaz svih knjiga. Mogućnost sortiranja po ceni ili imenu knjige u rastućem i opadajućem poretку. Mogućnost filtriranja po žanru, pretraga po imenu knjige ili imenu autora. Knjige dohvaćene AJAX-om i ispisane kroz JavaScript, filteri i sortiranje se takođe šalju JavaScriptom. Napravljena paginacija kroz JavaScript i usklađena s filtriranjem i sortiranjem tako da se zadrže izabrane vrednosti prilikom promene stranice. Klikom na dugme Add to cart proizvod se dodaje u korpu ako je korisnik ulogovan i dobija Toastr poruku da je uspešno dodato. Ako je admin dobija poruku da ne sme da dodaje, ako je gost, odnosno niko nije ulogovan dobija poruku da mora prvo da se uloguje. Klikom na ime knjige otvara se strana za prikaz informacija o jednoj knjizi. Ukoliko nema proizvoda koji odgovaraju kriterijumima pretrage prikazuje se poruka o tome. Paginacija prikazuje 9 proizvoda po strani.

Sort by:

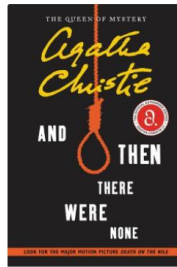
Price asc

Search

Genres:

☐ Horror (7)☐ Kids (6)☐ Mystery (6)☐ Poetry (6)☐ Romance (7)☐ Sci-Fi & Fantasy (11)☐ Thriller (6)

Books (49)



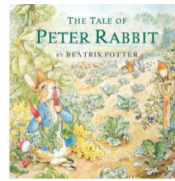
And Then There Were None

Genre: Mystery

Author: Agatha Christie

Price: \$3

Add to cart



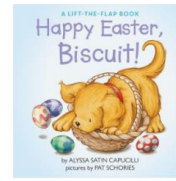
The Tale of Peter Rabbit

Genre: Kids

Author: Beatrix Potter

Price: \$5

Add to cart



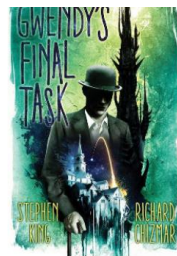
Happy Easter, Biscuit!

Genre: Kids

Author: Alyssa Satin Capucilli

Price: \$6

Add to cart



Gwendy's Final Task

Genre: Horror

Author: Stephen King,
Richard Chizmar

Price: \$9

Add to cart



Dune

Genre: Sci-Fi & Fantasy

Author: Frank Herbert

Price: \$9

Add to cart



Wingbearer

Genre: Kids

Author: Marjorie Liu

Price: \$9

Add to cart

1 2 3 4 5

Follow us

f Facebook
 Instagram
 YouTube

Working hours

🕒 Workdays: 08-20h
 🕒 Weekends: 09-18h
 📍 Knez Mihailova 1, Belgrade

Links

📖 Documentation
 👤 Author
 ©Djordje Solomun 16/19

Primer pokušaja dodavanja u korpu bez logovanja.

Sort by:

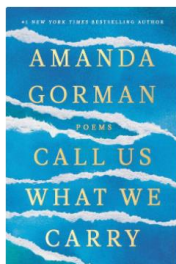
Price asc

Search

Genres:

- ☐ Horror (7)
- ☐ Kids (6)
- ☐ Mystery (6)
- ☐ Poetry (6)
- ☐ Romance (7)
- ☐ Sci-Fi & Fantasy (11)
- ☐ Thriller (6)

Books (49)



Call Us What We Carry

Genre: Poetry

Author: Amanda Gorman

Price: \$21

Add to cart

1 2 3 4 5

You have to login first.

Primer filtriranja i pretrage.

Sort by:

Price desc

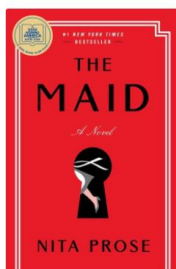
Search

the

Genres:

- ☐ Horror (7)
- ☐ Kids (6)
- ☒ Mystery (6)
- ☐ Poetry (6)
- ☐ Romance (7)
- ☐ Sci-Fi & Fantasy (11)
- ☐ Thriller (6)

Books (2)



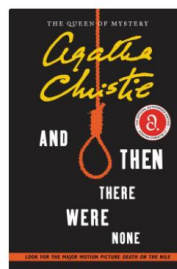
The Maid

Genre: Mystery

Author: Nita Prose

Price: \$13

Add to cart



And Then There Were None

Genre: Mystery

Author: Agatha Christie

Price: \$3

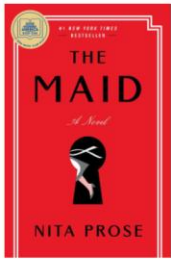
Add to cart

1

2.3.3 Stranica za prikaz jedne knjige

Nema posebnih funkcionalnosti, dodavanje u korpu isto kao na Shop stranici. Prikaz svih podataka o knjizi.

[Bookstore](#) [Home](#) [Shop](#) [Contact](#) [Login](#) [Register](#)



The Maid

Author: Nita Prose

Genre: Mystery

Publication year: 2022

Description:

Molly Gray is not like everyone else. She struggles with social skills and misreads the intentions of others. Her gran used to interpret the world for her, codifying it into simple rules that Molly could live by. Since Gran died a few months ago, twenty-five-year-old Molly has been navigating life's complexities all by herself. No matter—she throws herself with gusto into her work as a hotel maid. Her unique character, along with her obsessive love of cleaning and proper etiquette, make her an ideal fit for the job. She delights in donning her crisp uniform each morning, stocking her cart with miniature soaps and bottles, and returning guest rooms at the Regency Grand Hotel to a state of perfection. But Molly's orderly life is upended the day she enters the suite of the infamous and wealthy Charles Black, only to find it in a state of disarray and Mr. Black himself dead in his bed. Before she knows what's happening, Molly's unusual demeanor has the police targeting her as their lead suspect. She quickly finds herself caught in a web of deception, one she has no idea how to untangle. Fortunately for Molly, friends she never knew she had unite with her in a search for clues to what really happened to Mr. Black—but will they be able to find the real killer before it's too late? A Clue-like, locked-room mystery and a heartwarming journey of the spirit, *The Maid* explores what it means to be the same as everyone else and yet entirely different—and reveals that all mysteries can be solved through connection to the human heart.

Price: \$13

[Add to cart](#)

Follow us

- Facebook
- Instagram
- YouTube

Working hours

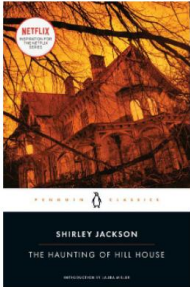
- Workdays: 08-20h
- Weekends: 09-18h
- Knez Mihailova 1, Belgrade

Links

- Documentation
- Author
- ©Djordje Solomun 16/19

Ovako admin vidi stranicu za prikaz jedne knjige:

[Bookstore](#) [Home](#) [Shop](#) [Contact](#) [Admin panel](#) Marko Markovic [Logout](#)



The Haunting of Hill House

Author: Shirley Jackson

Genre: Horror

Publication year: 2006

Description:

First published in 1959, Shirley Jackson's *The Haunting of Hill House* has been hailed as a perfect work of unnerving terror. It is the story of four seekers who arrive at a notoriously unfriendly pile called Hill House: Dr. Montague, an occult scholar looking for solid evidence of a "haunting"; Theodora, his lighthearted assistant; Eleanor, a friendless, fragile young woman well acquainted with poltergeists; and Luke, the future heir of Hill House. At first, their stay seems destined to be merely a spooky encounter with inexplicable phenomena. But Hill House is gathering its powers—and soon it will choose one of them to make its own.

Price: \$14

[Edit](#)
[Delete](#)

Follow us

- Facebook
- Instagram
- YouTube

Working hours

- Workdays: 08-20h
- Weekends: 09-18h
- Knez Mihailova 1, Belgrade

Links

- Documentation
- Author
- ©Djordje Solomun 16/19

2.3.4 Contact

Osnovne kontakt informacije i forma za slanje poruke validirana i na klijentskoj i na serverskoj strani. Ako je neki podatak pogrešno unet ispod polja se ispisuje adekvatna poruka. Ako je sve ispravno uneto i poruka poslata ispisaće se poruka o uspehu.

BookstoreHomeShopContactLoginRegister

Contact us

Send us a message

Full name *

ex. John Doe

Email *

ex. johndoe@gmail.com

Your message *

Your message.

* Required fields

Send

Our contact information

Knez Mihailova 10, Belgrade

+381 64 987654

+381 11 234567

bookstore@gmail.com

Follow usFacebook

Working hoursWorkdays: 08-20h

LinksDocumentation

BookstoreHomeShopContactLoginRegister

Contact us

Send us a message

Full name *

ex. John Doe

First and last name must start with capital letter and have at least 2 characters.

Email *

ex. johndoe@gmail.com

Email field can't be empty.

Your message *

Your message.

Please enter a message no longer than 200 characters.

* Required fields

Send

Our contact information

Knez Mihailova 10, Belgrade

+381 64 987654

+381 11 234567

bookstore@gmail.com

14

[Bookstore](#) [Home](#) [Shop](#) [Contact](#) [Login](#) [Register](#)

Contact us

Send us a message

Full name *

Email *


Your message *


* Required fields


Message sent!


Send

Our contact information


 Knez Mihailova 10, Belgrade

 +381 64 987654


 +381 11 234567

 bookstore@gmail.com


Follow us

 Facebook

Working hours

 Workdays: 08-20h

Links

 Documentation

2.3.5 Register

Forma za registraciju, validirana i sa klijentske i sa serverske strane. Ako je neki podatak pogrešno unet ispod polja se ispisuje adekvatna poruka. Ako je sve ispravno uneto ispisuje se poruka o uspešnoj registraciji i link koji vodi na login stranu. Ulogovani korisnici ne mogu opet doći na stranicu za registraciju.

[Bookstore](#) [Home](#) [Shop](#) [Contact](#) [Login](#) [Register](#)

Register

Full name *




Email *

Password *




* Required fields

Register



Follow us

 Facebook
 Instagram
 YouTube

Working hours

 Workdays: 08-20h
 Weekends: 09-18h
 Knez Mihailova 1, Belgrade

Links

 Documentation
 Author
©Djordje Solomun 16/19

[Bookstore](#) [Home](#) [Shop](#) [Contact](#) [Login](#) [Register](#)

Register

Registered successfully.
Click [here](#) to login now

Full name *

Email *

Password *

* Required fields

Register

Follow us

Working hours

Links

[Facebook](#)

⌚ Workdays: 08-20h

[Documentation](#)

2.3.6 Login

Forma za login. Validirana i sa klijentske i sa serverske strane. Ako je neki podatak pogrešno unet ispod polja se ispisuje adekvatna poruka. Ako su email ili lozinka pogrešni ispisaće se poruka o tome. Ako je login uspešan korisnici se vraćaju na home, a admin se šalje na admin panel.

[Bookstore](#) [Home](#) [Shop](#) [Contact](#) [Login](#) [Register](#)

Login

Email

Password

Login

Follow us

Working hours

Links

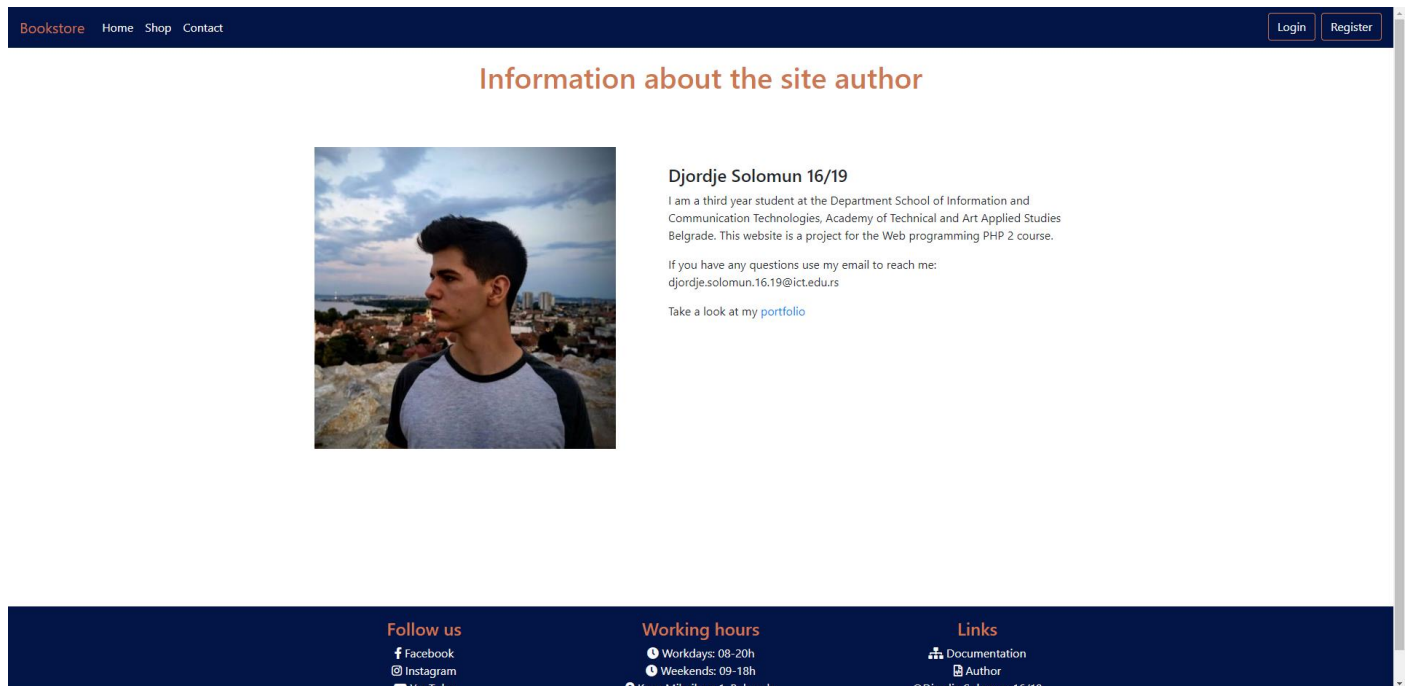
[Facebook](#)
[Instagram](#)
[YouTube](#)

⌚ Workdays: 08-20h
⌚ Weekends: 09-18h
📍 Knez Mihailova 1, Belgrade

[Documentation](#)
👤 Author
© Djordje Solomun 16/19

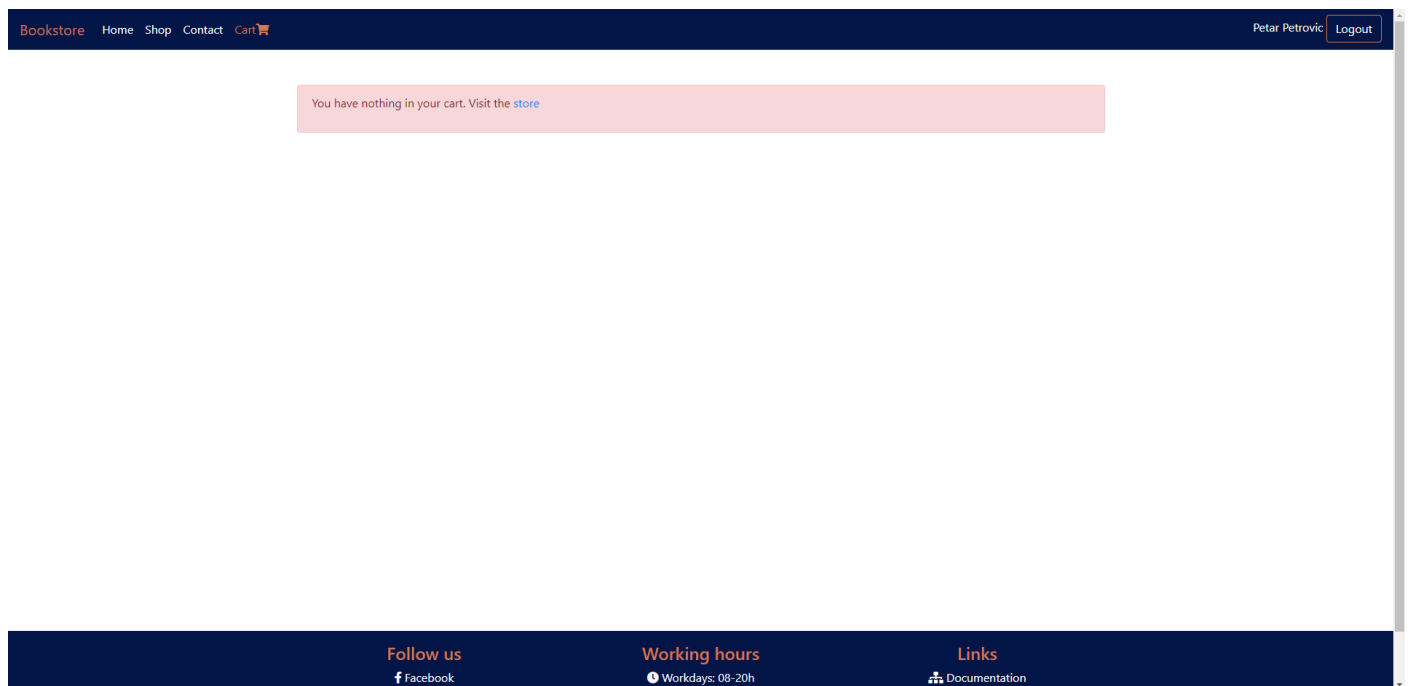
2.3.7 Author

Nema posebnih funkcionalnosti. Osnovne informacije o autoru sajta.



2.3.8 Cart

Prikaz svih proizvoda u korisnikovoj korpi, njihova količina i ukupna cena svega u korpi. Ovo je inicijalni prikaz odnosno kada je korpa prazna.



Prikaz kada nešto postoji u korpi. Korpa i njen sadržaj se čuvaju u sesiji i odatle se ispisuju, a funkcionalnost brisanja iz korpe i menjanja količine se postiže AJAX zahtevima i JavaScript ažurira prikaz.

[Bookstore](#) [Home](#) [Shop](#) [Contact](#) [Cart](#)

Petar Petrovic [Logout](#)

Shopping Cart

3 items

	Book Happy Easter, Biscuit!	Quantity <input type="text" value="1"/>	Price €6	×
	Book The Tale of Peter Rabbit	Quantity <input type="text" value="1"/>	Price €5	×
	Book The Lorax	Quantity <input type="text" value="1"/>	Price €8	×

[← Back to shop](#)

Total price: € 19
[Checkout](#)

Kada se obriše poslednji proizvod prikaz će se osvežiti i ponovo će pisati poruka da je korpa prazna. Klikom na dugme Checkout korpa će da se isprazni iz sesije i korisnik će dobiti sledeći prikaz.

[Bookstore](#) [Home](#) [Shop](#) [Contact](#) [Cart](#)

Petar Petrovic [Logout](#)

Your order has been placed. Go back to the [store](#)

Follow us
[f](#) Facebook

Working hours
[🕒](#) Workdays: 09-20h

Links
[📖](#) Documentation

2.3.9 Admin home

Početna stranica za admina na njoj se nalazi prikaz svih ključnih aktivnosti od strane korisnika u vidu tabele. Urađena paginacija pomoću Laravela, mogućnost sortiranja po datumu u rastućem i opadajućem poretku.

[Bookstore](#) [Admin home](#) [Admin products](#) [Admin genres](#) [Admin users](#) [Admin messages](#) Marko Markovic [Logout](#)

Admin panel

Recent user activity

Sort by Date ASC [Sort](#)

Activity	User name	Date
Login	Petar Petrovic	2022-03-14 19:40:16
Logged out	Petar Petrovic	2022-03-14 19:40:58
Login	Marko Markovic	2022-03-14 19:41:08
Logged out	Marko Markovic	2022-03-14 19:46:26
Login	Marko Markovic	2022-03-14 19:56:55
Logged out	Marko Markovic	2022-03-14 19:57:37
Sent a message	John Doe	2022-03-14 20:00:06
Registered	Pera Peric	2022-03-14 20:02:32
Login	Petar Petrovic	2022-03-14 20:10:18

< 1 2 >

Follow us

- Facebook
- Instagram
- YouTube

Working hours

- Workdays: 08-20h
- Weekends: 09-18h
- Knez Mihailova 1, Belgrade

Links

- Documentation
- Author
- ©Djordje Solomun 16/19

2.3.10 Admin products

Prikaz svih proizvoda sa filtriranjem i sortiranjem. Prikaz, filtriranje i sortiranje su odrađeni kroz Laravel. Dugme za dodavanje novog proizvoda, dugme za edit i brisanje nekog od proizvoda.

[Bookstore](#) [Admin home](#) [Admin products](#) [Admin genres](#) [Admin users](#) [Admin messages](#) Marko Markovic [Logout](#)

Sort by:

Price ASC

Search

Genres:

☐ Horror (7)

☐ Kids (6)

☐ Mystery (6)

☐ Poetry (6)

☐ Romance (7)

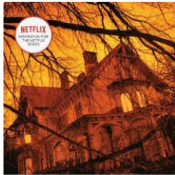
☐ Sci-Fi & Fantasy (11)

☐ Thriller (6)

Search

Books (49)

[Add a new book](#)




The Haunting of Hill House

Genre: Horror

Author: Shirley Jackson

Price: \$14

[Edit](#)[Delete](#)



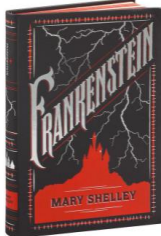
Gwendy's Final Task

Genre: Horror

Author: Stephen King, Richard Chizmar

Price: \$9

[Edit](#)[Delete](#)



Frankenstein

Genre: Horror

Author: Mary Shelley

Price: \$15

[Edit](#)[Delete](#)

19

2.3.11 Add new book

Forma za unos novog proizvoda. Validirana kroz laravel, sva polja su obavezna i ispod svakog će se ispisati opisna greška ako je nešto pogrešno uneto. Drugi autor nije obavezan. Ako se proizvod uspešno unese admin se šalje na stranicu za prikaz svih proizvoda i tamo dobija poruku o uspešnom unosu. Uneti autor će pokušati da se pronade u bazi, ukoliko postoji knjiga će se spojiti sa njim, a ako ne postoji biće unet novi autor sa tim imenom i spojiće se sa knjigom.

The screenshot shows the 'Add a new book' form in a web application. The form is titled 'Add a new book' in orange. It contains several input fields: 'Title' (with placeholder 'Book title'), 'Select genre' (a dropdown menu with 'Select' as the current selection), 'Author' (with a sub-field 'Optional'), 'Price', 'Publication year', and 'Description' (a large text area). Below these fields is an 'Image' section with a 'Choose File' button and the text 'No file chosen'. At the bottom right of the form is an 'Add' button. The form is set against a dark blue header with navigation links: 'Bookstore', 'Admin home', 'Admin products', 'Admin genres', 'Admin users', and 'Admin messages'. On the right of the header, it says 'Marko Markovic' and 'Logout'.

2.3.12 Edit book

Forma za edit postojećeg proizvoda, već popunjena podacima o tom proizvodu, sva polja osim slike su obavezna. Ako se pošalje i slika stara će biti obrisana, nova upload-ovana i biće promenjen naziv slike u bazi.

The screenshot shows the 'Edit book' form in a web application. The form is titled 'Edit book' in orange. It contains several input fields: 'Title' (with placeholder 'The Haunting of Hill House'), 'Select genre' (a dropdown menu with 'Horror' as the current selection), 'Author' (with placeholder 'Shirley Jackson'), 'Price' (with placeholder '14'), 'Publication year' (with placeholder '2006'), and 'Description' (a large text area with placeholder text). Below these fields is an 'Image' section with a 'Choose File' button and the text 'No file chosen'. At the bottom right of the form is an 'Edit' button. The form is set against a dark blue header with navigation links: 'Bookstore', 'Admin home', 'Admin products', 'Admin genres', 'Admin users', and 'Admin messages'. On the right of the header, it says 'Marko Markovic' and 'Logout'.

2.3.13 Admin genres

Prikaz svih žanrova u vidu tabele. Dugme za dodavanje novog žanra, dugme za edit postojećeg i dugme za brisanje. Brisanje neće biti dozvoljeno ako neke knjige pripadaju tom žanru. Ako se žanr uspešno obriše na ovoj stranici će biti prikazana poruka o uspehu.

[Bookstore](#) [Admin home](#) [Admin products](#) [Admin genres](#) [Admin users](#) [Admin messages](#) Marko Markovic [Logout](#)

Genres

[Add new genre](#)

Name	Edit	Delete
Horror	Edit	Delete
Kids	Edit	Delete
Mystery	Edit	Delete
Poetry	Edit	Delete
Romance	Edit	Delete
Sci-Fi & Fantasy	Edit	Delete
Thriller	Edit	Delete

Follow us
[Facebook](#)
[Instagram](#)
[YouTube](#)

Working hours
🕒 Workdays: 08-20h
🕒 Weekends: 09-18h
📍 Knez Mihailova 1, Belgrade

Links
📖 Documentation
👤 Author
©Djordje Solomun 16/19

2.3.14 Add new genre

Forma za dodavanje novog žanra, ime žanra ne sme da bude ime već postojećeg žanra. Ako je unos uspešan povratak na stranicu sa svim žanrovima sa porukom o uspehu.

[Bookstore](#) [Admin home](#) [Admin products](#) [Admin genres](#) [Admin users](#) [Admin messages](#) Marko Markovic [Logout](#)

Add a new genre

Genre name

[Add](#)

2.3.15 Edit genre

Forma za edit žanra popunjena podacima o žanru koji se edituje. Sme da se ostavi isto ime tada se neće validirati da bude jedinstveno u tabeli za žanrove, ali ukoliko se promeni tada se primenjuje validacija i ne sme se uneti ime nekog od postojećih žanrova. Ako je edit uspešan povratak na stranicu sa svim žanrovima sa porukom o uspehu.

[Bookstore](#) [Admin home](#) [Admin products](#) [Admin genres](#) [Admin users](#) [Admin messages](#) Marko Markovic [Logout](#)

Edit genre

Genre name

Edit

Follow us

- Facebook
- Instagram
- YouTube

Working hours

- Workdays: 08-20h
- Weekends: 09-18h
- Knez Mihailova 1, Belgrade

Links

- Documentation
- Author
- ©Djordje Solomun 16/19

2.3.16 Admin users

Prikaz svih korisnika u vidu tabele. Dugme za brisanje korisnika. Kad je brisanje uspešno na ovoj stranici se prikazuje poruka o uspehu.

[Bookstore](#) [Admin home](#) [Admin products](#) [Admin genres](#) [Admin users](#) [Admin messages](#) Marko Markovic [Logout](#)

Users

User ID	Name	Email	Role	Delete
1	Petar Petrovic	pera@gmail.com	user	Delete
11	Laza Lazic	laza@gmail.com	user	Delete
31	Pera Peric	petar@gmail.com	user	Delete

Follow us

- Facebook
- Instagram
- YouTube

Working hours

- Workdays: 08-20h
- Weekends: 09-18h
- Knez Mihailova 1, Belgrade

Links

- Documentation
- Author
- ©Djordje Solomun 16/19

3. Kodovi

3.1 Kontroleri

3.1.1 AdminController

```
<?php

namespace App\Http\Controllers;

use App\Models\Activity;
use App\Models\Book;
use App\Models\Contact;
use App\Models\Genre;
use App\Models\User;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Log;

class AdminController extends BaseController
{
    public function index(Request $request){
        $qs=[];
        $activities = new Activity();
        if($request->has('sortBy')){
            $qs['sort'] = $request->get('sortBy');
            $sort=explode('-', $request->get('sortBy'));
            $activities = $activities->orderBy($sort[0], $sort[1]);
        }
        $this->data['qs'] = $qs;
        $this->data['activities'] = $activities->paginate(9)->withQueryString();
        return view('pages.admin.index', $this->data);
    }

    public function products(Request $request){
        $genres = Genre::with('books')->get();
        $books = Book::with('authors')->with('genre');
        $qs = [];

        if($request->has('chbGenres') && $request->get('chbGenres')){
            $qs['chbGenres'] = $request->chbGenres;
            $books = $books->whereIn("genre_id", $request->get("chbGenres"));
        }

        if($request->has('search') && $request->get('search')){
            $search = $request->search;
            $qs['search'] = $search;
            $books = $books->where(function($q) use ($search){

                $q->whereHas("authors", function($q) use ($search){
                    $q->where('name', 'like', "%".$search."%");
                })
                ->orWhere('book_name', 'like', "%".$search."%");
            });
        }

        if($request->has('sortBy') && $request->get('sortBy')){
            $qs['sortBy'] = $request->get('sortBy');
            $sort = explode("-", $request->get("sortBy"));

            $books = $books->orderBy($sort[0], $sort[1]);
        }
        $this->data['qs'] = $qs;
        $this->data['genres'] = $genres;
        $this->data['books'] = $books->paginate(12)->withQueryString();
        return view('pages.admin.books.products', $this->data);
    }
}
```

```

    }

    public function users(){
        $this->data['users'] = User::with('role')->get();
        return view('pages.admin.users.index',$this->data);
    }

    public function destroy_user($id){
        $user = User::find($id);
        try{
            $user->delete();
            Activity::create([
                'type'=>'Deleted a user',
                'user_name'=>session()->get('user')->name
            ]);
            return redirect()->back()->with('success','User deleted!');
        }catch(\Exception $e){
            Log::error($e->getMessage());
            return redirect()->back()->with('error','There was an error processing the request.');
```

3.1.2 AuthController

```

<?php

namespace App\Http\Controllers;

use App\Http\Requests\LoginRequest;
use App\Http\Requests\RegisterRequest;
use App\Models\Activity;
use App\Models\User;
use Illuminate\Support\Facades\Log;

class AuthController extends BaseController
{
    public function loginForm(){
        return view('pages.auth.login');
    }

    public function registerForm(){
        return view('pages.auth.register');
    }

    public function login(LoginRequest $request){
        $email = trim($request->tbEmail);
        $pass = md5($request->tbPass);

        $user = User::where('email',$email)
            ->where('password',$pass)
            ->with("role")
            ->first();
        if(!$user){
            return redirect()->back()->with("error","Wrong username or password.");
        }
        Activity::create([
```



```

        'type'=>'Login',
        'user_name'=>$user->name
    ]);
    session()->put("user",$user);
    if($user->role->role_name == 'admin'){
        return redirect()->route('admin.home');
    }
    else{
        return redirect()->route("home");
    }
}

public function register(RegisterRequest $request){
    $name = $request->tbName;
    $email = $request->tbEmail;
    $password = md5($request->tbPass);

    $user = new User;

    $user->name = $name;
    $user->email = $email;
    $user->password = $password;
    try{
        $res = $user->save();
        if($res){
            Activity::create([
                'type'=>'Registered',
                'user_name'=>$user->name
            ]);
            return redirect()->back()->with('success','Registered successfully.');
```

```

        }
    }catch(\Exception $e){
        Log::error($e->getMessage());
        return redirect()->back()->with('error','There was an error processing your
request.');
```

```

    }
}
```

```

public function logout(){
    Activity::create([
        'type'=>'Logged out',
        'user_name'=>session()->get('user')->name
    ]);
    session()->forget('user');

    return redirect()->route('home');
}
}
```

3.1.3 BaseController

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class BaseController extends Controller
{
    public $data;

    public function __construct(){
```

```
}  
}
```

3.1.4 BookController

```
<?php  
  
namespace App\Http\Controllers;  
  
use App\Http\Requests\BookStoreRequest;  
use App\Http\Requests\BookUpdateRequest;  
use App\Models\Activity;  
use App\Models\Author;  
use App\Models\Book;  
use App\Models\Genre;  
use Illuminate\Http\Request;  
use Illuminate\Support\Facades\File;  
use Illuminate\Support\Facades\Log;  
  
class BookController extends BaseController  
{  
  
    public function __construct(){  
        parent::__construct();  
        $this->middleware('isAdmin')  
            ->only([  
                'destroy',  
                'store',  
                'update',  
                'create',  
                'edit'  
            ]);  
    }  
  
    public function get_json(Request $request){  
        if($request->ajax()){  
            $books = Book::with('genre')->with('authors');  
  
            if($request->genres){  
                $books = $books->whereIn('genre_id',$request->genres);  
            }  
  
            if($request->search && !empty($request->search)){  
                $search = $request->search;  
                $books = $books->where(function($q) use ($search){  
                    $q->whereHas("authors", function($q) use ($search){  
                        $q->where('name','like',"%".$search."%");  
                    })  
                    ->orWhere('book_name','like',"%".$search."%");  
                });  
            }  
  
            if($request->sort){  
                $column = explode('-', $request->sort)[0];  
                $type = explode('-', $request->sort)[1];  
                $books = $books->orderBy($column,$type);  
            }  
  
            header('Content-type:application/json');  
            echo json_encode($books->paginate(12));  
        }  
        else{  
            return redirect()->back();  
        }  
    }  
}
```

```

    }
}
public function index()
{
    $genres = Genre::with('books')->get();
    $this->data['genres']=$genres;
    return view('pages.books.index',$this->data);
}

public function create()
{
    $genres = Genre::all();
    $this->data['genres']=$genres;
    return view('pages.admin.books.create',$this->data);
}

public function store(BookStoreRequest $request)
{
    $title = $request->get('title');
    $author_1 = $request->get('author1');
    if(isset($request->author2)){
        $author_2 = $request->get('author2');
    }
    $price = $request->get('price');
    $publishYear = $request->get('publishYear');
    $description = $request->get('description');
    $genreId = $request->get('genreId');
    $extension = $request->image->extension();
    $imageName = strtolower(implode('_',explode(
',str_replace(':', '', $title)))).'.'.$extension;
    $request->image->move(public_path('img/books'), $imageName);

    try{
        $book = Book::create([
            'book_name'=>$title,
            'price' => $price,
            'description'=>$description,
            'image'=>$imageName,
            'publish_year'=>$publishYear,
            'genre_id'=>$genreId
        ]);
        $authors = [];
        $author = Author::firstOrCreate(['name'=>$author_1]);
        array_push($authors,$author->id);
        if(isset($author_2)){
            $author2 = Author::firstOrCreate(['name'=>$author_2]);
            array_push($authors,$author2->id);
        }
        $book->authors()->sync($authors);
        Activity::create([
            'type'=>'Added new book',
            'user_name'=>session()->get('user')->name
        ]);
        return redirect()->route('admin.products')->with('success','true');
    }catch(\Exception $e){
        Log::error($e->getMessage());
        return redirect()->back()->with("error", $e->getMessage());
    }
}

public function show($id)

```

```

{
    $this->data['book'] = Book::with('authors')->with('genre')->find($id);
    return view('pages.books.show',$this->data);
}

public function edit($id)
{
    $genres = Genre::all();
    $this->data['genres']=$genres;
    $this->data['book'] = Book::with('authors')->with('genre')->find($id);
    return view('pages.admin.books.edit',$this->data);
}

public function update(BookUpdateRequest $request, $id)
{
    $title = $request->get('title');
    $author_1 = $request->get('author1');
    if(isset($request->author2)) {
        $author_2 = $request->get('author2');
    }
    $price = $request->get('price');
    $publishYear = $request->get('publishYear');
    $description = $request->get('description');
    $genreId = $request->get('genreId');

    try{
        $book = Book::with('authors')->with('genre')->find($id);
        $authors = [];
        $author = Author::firstOrCreate(['name'=>$author_1]);
        array_push($authors,$author->id);
        if(isset($author_2)) {
            $author2 = Author::firstOrCreate(['name'=>$author_2]);
            array_push($authors,$author2->id);
        }
        $book->book_name = $title;
        $book->price = $price;
        $book->publish_year = $publishYear;
        $book->genre_id = $genreId;
        $book->description = $description;
        if(isset($request->image)) {
            $oldImage = public_path("img/books/{$book->image}");
            if (File::exists($oldImage)) {
                unlink($oldImage);
            }
            $extension = $request->image->extension();
            $imageName = strtolower(implode('_',explode(' ',str_replace(':', '', $title)))).'.'.$extension;
            $request->image->move(public_path('img/books'), $imageName);
            $book->image = $imageName;
        }
        $book->save();
        $book->authors()->sync($authors);
        Activity::create([
            'type'=>'Updated book',
            'user_name'=>session()->get('user')->name
        ]);
        return redirect()->route('admin.products')->with('success','Updated book!');
    }catch(\Exception $e){
        Log::error($e->getMessage());
        return redirect()->back()->with("error", $e->getMessage());
    }
}

```

```

}

public function destroy($id)
{
    $book = Book::find($id);
    $image_path = public_path("img/books/{$book->image}");
    if (File::exists($image_path)) {
        unlink($image_path);
    }
    try{
        $book->authors()->detach();
        $book->delete();
        Activity::create([
            'type'=>'Deleted a book',
            'user_name'=>session()->get('user')->name
        ]);
        return redirect()->route('admin.products')->with('success','Book
deleted!');
    }catch(\Exception $e){
        Log::error($e->getMessage());
        return redirect()->route('admin.products')->with('error',"There was an
error processing the request.");
    }
}
}

```

3.1.5 CartController

```

<?php

namespace App\Http\Controllers;

use App\Models\Activity;
use App\Models\Book;
use Illuminate\Http\Request;

class CartController extends BaseController
{
    public function index(Request $request) {

        if(session()->has('user') && session()->get('user')->role->role_name ==
'user'){

            try {
                $bookId = $request->get("bookId");

                if(!$request->session()->has("cart")) {
                    $request->session()->put("cart", []);
                }

                $book = Book::find($bookId);
                $cart = $request->session()->get("cart");

                if(isset($cart[$bookId])) {
                    $alreadyInCart = $cart[$bookId];

                    $alreadyInCart->quantity++;

                    $cart[$bookId] = $alreadyInCart;
                } else {
                    $item = new \stdClass();
                    $item->bookId = $bookId;
                    $item->quantity = 1;
                    $item->price = $book->price;

```

```

        $item->image = $book->image;
        $item->name = $book->book_name;
        $cart[$bookId] = $item;
    }

    $request->session()->put("cart", $cart);
    Activity::create([
        'type'=>'Added item to cart',
        'user_name'=>session()->get('user')->name
    ]);
    echo "uspeh";

    } catch(\Exception $e) {
        echo "greska";
    }
}

elseif(session()->has('user') && session()->get('user')->role-
>role_name == 'admin'){
    echo 'adminje';
}
else{
    echo 'gostje';
}

}

public function adjustQuantity(Request $request) {
    if(session()->get('user')->role->role_name == 'admin'){
        return redirect()->back();
    }
    $bookId = $request->get("bookId");
    $quantity = $request->get("quantity");

    $cart = $request->session()->get("cart");

    if(isset($cart[$bookId])) {
        $alreadyInCart = $cart[$bookId];

        $alreadyInCart->quantity = $quantity;

        $cart[$bookId] = $alreadyInCart;

        $request->session()->put("cart", $cart);
    }
}

public function get(Request $request) {
    if(session()->get('user')->role->role_name == 'admin'){
        return redirect()->back();
    }
    $cart = $request->session()->get("cart");
    if(!$cart) {
        $cart = [];
    }

    return view("pages.books.cart", [
        "cartItems" => $cart ]);
}

public function remove($bookId, Request $request) {
    if(session()->get('user')->role->role_name == 'admin'){
        return redirect()->back();
    }
    $cart = $request->session()->get("cart");

```

```

        if(isset($cart[$bookId])) {
            unset($cart[$bookId]);
            Activity::create([
                'type'=>'Removed item from cart',
                'user_name'=>session()->get('user')->name
            ]);
            $request->session()->put("cart", $cart);
        }
    }

    public function checkout() {
        if(session()->get('user')->role->role_name == 'admin'){
            return redirect()->back();
        }
        Activity::create([
            'type'=>'Checked out',
            'user_name'=>session()->get('user')->name
        ]);
        session()->forget('cart');
        return redirect()->back()->with('msg','true');
    }
}

```

3.1.6 ContactController

```

<?php

namespace App\Http\Controllers;

use App\Http\Requests\ContactRequest;
use App\Models\Activity;
use App\Models>Contact;
use Illuminate\Support\Facades\Log;

class ContactController extends BaseController
{
    public function index() {
        return view('pages.contact');
    }

    public function send(ContactRequest $request) {
        $name = $request->tbName;
        $email = $request->tbEmail;
        $message = $request->userMessage;
        try{
            Contact::create([
                'name' => $name,
                'email' => $email,
                'message' => $message
            ]);
            Activity::create([
                'type' => 'Sent a message',
                'user_name' => $name
            ]);
            return redirect()->back()->with('success', 'Message sent!');
        }
        catch(\Exception $e) {
            Log::error($e->getMessage());
            return redirect()->back()->with('error','There was an error sending your
message');
        }
    }
}

```

```
}
```

3.1.7 Controller (Laravelov kontroler nije menjan)

```
<?php

namespace App\Http\Controllers;

use Illuminate\Foundation\Auth\Access\AuthorizesRequests;
use Illuminate\Foundation\Bus\DispatchesJobs;
use Illuminate\Foundation\Validation\ValidatesRequests;
use Illuminate\Routing\Controller as BaseController;

class Controller extends BaseController
{
    use AuthorizesRequests, DispatchesJobs, ValidatesRequests;
}
```

3.1.8 GenreController

```
<?php

namespace App\Http\Controllers;

use App\Http\Requests\GenreStoreRequest;
use App\Http\Requests\GenreUpdateRequest;
use App\Models\Activity;
use App\Models\Genre;
use Illuminate\Support\Facades\Log;

class GenreController extends BaseController
{
    public function index() {
        $genres = Genre::with('books')->get();
        $this->data['genres'] = $genres;
        return view('pages.admin.genres.index', $this->data);
    }

    public function create() {
        return view('pages.admin.genres.create');
    }

    public function store(GenreStoreRequest $request) {
        $name = $request->genreName;
        try{
            Genre::create([
                'genre_name'=>$name
            ]);
            Activity::create([
                'type'=>'Added a new genre',
                'user_name'=>session()->get('user')->name
            ]);
            return redirect()->route('admin.genres')->with('success', "New genre
added!");
        } catch (\Exception $e) {
            Log::error($e->getMessage());
            return redirect()->back()->with('error', "There was an error processing the
request.");
        }
    }
}
```



```

public function edit($id) {
    $this->data['genre'] = Genre::find($id);
    return view('pages.admin.genres.edit', $this->data);
}

public function update(GenreUpdateRequest $request, $id) {
    $genre = Genre::find($id);
    $name = $request->genreName;
    if($genre->genre_name != $name) {
        $request->validate(['genreName' => 'unique:genres,genre_name']);
    }
    try{
        $genre->genre_name = $name;
        $genre->save();
        Activity::create([
            'type'=>'Updated a genre',
            'user_name'=>session()->get('user')->name
        ]);
        return redirect()->route('admin.genres')->with('success','Updated genre');
    }catch(\Exception $e) {
        Log::error($e->getMessage());
        return redirect()->back()->with('error','There was an error processing your
request.');
```

```

    }
}

public function destroy($id) {
    $genre = Genre::with('books')->find($id);
    if(count($genre->books) > 0) {
        return redirect()->back()->with('error',"Genre has books associated with
it, can't delete.");
    }
    try{
        $genre->delete();
        Activity::create([
            'type'=>'Deleted a genre',
            'user_name'=>session()->get('user')->name
        ]);
        return redirect()->back()->with('success',"Genre deleted!");
    }catch(\Exception $e) {
        Log::error($e->getMessage());
        return redirect()->back()->with('error',"There was an error processing the
request.');
```

```

    }
}
}

```

3.1.9 HomeController

```

<?php

namespace App\Http\Controllers;

use App\Models\Book;

class HomeController extends BaseController
{
    public function index() {
        $books = Book::with('authors')->with('genre')->orderBy('created_at', 'desc')-
>take(4)->get();
        $this->data['books'] = $books;
        return view('pages.home', $this->data);
    }
}

```

```

public function author() {
    return view('pages.author');
}
}

```

3.2. Middleware

3.2.1 IsAdmin

```

<?php

namespace App\Http\Middleware;

use Closure;
use Illuminate\Http\Request;

class IsAdmin
{
    /**
     * Handle an incoming request.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  \Closure(\Illuminate\Http\Request):
     * (\Illuminate\Http\Response|\Illuminate\Http\RedirectResponse)  $next
     * @return \Illuminate\Http\Response|\Illuminate\Http\RedirectResponse
     */
    public function handle(Request $request, Closure $next)
    {
        if(session()->has('user') && session()->get('user')->role->role_name ==
'admin'){
            return $next($request);
        }
        return redirect()->back();
    }
}

```

3.2.2 IsGuest

```

<?php

namespace App\Http\Middleware;

use Closure;
use Illuminate\Http\Request;

class IsGuest
{
    /**
     * Handle an incoming request.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  \Closure(\Illuminate\Http\Request):
     * (\Illuminate\Http\Response|\Illuminate\Http\RedirectResponse)  $next
     * @return \Illuminate\Http\Response|\Illuminate\Http\RedirectResponse
     */
    public function handle(Request $request, Closure $next)
    {
        if(session()->get('user')){
            return redirect()->back();
        }
        return $next($request);
    }
}

```

3.2.3 IsLoggedIn

```
<?php

namespace App\Http\Middleware;

use Closure;
use Illuminate\Http\Request;

class isLoggedIn
{
    /**
     * Handle an incoming request.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  \Closure(\Illuminate\Http\Request): (\Illuminate\Http\Response|\Illuminate\Http\RedirectResponse)  $next
     * @return \Illuminate\Http\Response|\Illuminate\Http\RedirectResponse
     */
    public function handle(Request $request, Closure $next)
    {
        if(session()->get('user')){
            return $next($request);
        }
        return redirect()->route('login')->with('error','You have to login first!');
    }
}
```

3.3 Requests

3.3.1 BookStoreRequest

```
<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class BookStoreRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            'title'=>'required|string|min:4',
            'author1'=>'required|string|min:4',
            'author2'=>'nullable|string|min:4',
            'description'=>'required|string|min:10',
            'image'=>'required|image|max:2048',
            'genreId'=>'required|exists:genres,id',
            'price'=>'required|numeric',
        ];
    }
}
```

```

        'publishYear'=>'required|numeric'
    ];
}
public function messages()
{
    return [
        'title.required'=>'Title field is required',
        'author1.required'=>'Author field is required',
        'description.required'=>'Description field is required',
        'image.required'=>'The image is required',
        'title.string'=>'Title field can contain only text',
        'author1.string'=>'Author field can contain only text',
        'author2.string'=>'Author field can contain only text',
        'author1.min'=>'Author must have at least 4 characters',
        'author2.min'=>'Author must have at least 4 characters',
        'image.image'=>'Uploaded file must be an image',
        'image.max'=>'The image must be under 2MB',
        'genreId.required'=>'Genre is required',
        'price.required'=>'Price is required',
        'publishYear.required'=>'Publication year is required',
        'price.numeric'=>'Price must be a number',
        'publishYear.numeric'=>'Publication year must be a number',
    ];
}
}

```

3.3.2 BookUpdateRequest

```

<?php
namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class BookUpdateRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            'title'=>'required|string|min:4',
            'author1'=>'required|string|min:4',
            'author2'=>'nullable|string|min:4',
            'description'=>'required|string|min:10',
            'image'=>'nullable|image|max:2048',
            'genreId'=>'required|exists:genres,id',
            'price'=>'required|numeric',
            'publishYear'=>'required|numeric'
        ];
    }
}

```

```

public function messages()
{
    return [
        'title.required'=>'Title field is required',
        'author1.required'=>'Author field is required',
        'description.required'=>'Description field is required',
        'title.string'=>'Title field can contain only text',
        'author1.string'=>'Author field can contain only text',
        'author2.string'=>'Author field can contain only text',
        'author1.min'=>'Author must have at least 4 characters',
        'author2.min'=>'Author must have at least 4 characters',
        'image.image'=>'Uploaded file must be an image',
        'image.max'=>'The image must be under 2MB',
        'genreId.required'=>'Genre is required',
        'genreId.exists'=>'Please choose a genre',
        'price.required'=>'Price is required',
        'publishYear.required'=>'Publication year is required',
        'price.numeric'=>'Price must be a number',
        'publishYear.numeric'=>'Publication year must be a number',
    ];
}
}

```

3.3.3 ContactRequest

```

<?php
namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class ContactRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            'tbName'=>'required|string|min:5',
            'tbEmail' => 'required|email',
            'userMessage' => 'required|string|min:1|max:200'
        ];
    }

    public function messages()
    {
        return [
            'tbName.required' => 'Name field is required',
            'tbEmail.required' => 'Email field is required',
            'userMessage.required' => 'Message field is required',
            'tbEmail.email' => 'Please enter a valid email(example:

```

```

yourname@gmail.com).',
    'userMessage.max' => 'Please enter a message no longer than 200
characters.',
    'userMessage.min' => "Message field can't be empty"
];
}
}

```

3.3.4 GenreStoreRequest

```

<?php
namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class GenreStoreRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            'genreName' => 'required|string|min:3|unique:genres,genre_name'
        ];
    }

    public function messages()
    {
        return [
            'genreName.required' => "Genre name is required.",
            'genreName.string' => "Genre name can only have text.",
            'genreName.min' => "Genre name must have at least 3 characters.",
            'genreName.unique' => "This genre already exists"
        ];
    }
}

```

3.3.5 GenreUpdateRequest

```

<?php
namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class GenreUpdateRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */

```

```

    */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            'genreName' => 'required|string|min:3'
        ];
    }

    public function messages()
    {
        return [
            'genreName.required' => "Genre name is required.",
            'genreName.string' => "Genre name can only have text.",
            'genreName.min' => "Genre name must have at least 3 characters.",
            'genreName.unique' => "This genre already exists"
        ];
    }
}

```

3.3.6 LoginRequest

```

<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class LoginRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            'tbEmail' => 'required',
            'tbPass' => 'required'
        ];
    }
}

```

3.3.7 RegisterRequest

```
<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class RegisterRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            'tbName' => 'required|string|min:5',
            'tbEmail' => 'required|email|unique:users,email',
            'tbPass' => 'required|string|min:7'
        ];
    }

    public function messages()
    {
        return [
            'tbName.required' => 'Name field is required',
            'tbEmail.required' => 'Email field is required',
            'tbEmail.unique' => 'This email is already taken.',
            'tbEmail.email' => 'Please enter a valid email(example:
yourname@gmail.com).',
            'tbPass.required' => 'Password field is required'
        ];
    }
}
```

3.4 Models

3.4.1 Activity

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Activity extends Model
{
    use HasFactory;
    public $fillable=['type','user_name'];
}
```


3.4.2 Author

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Author extends Model
{
    use HasFactory;

    public $guarded=['id,created_at,updated_at'];

    public function books() {
        return $this->hasMany(Book::class)->withTimestamps();
    }
}
```

3.4.3 Book

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Book extends Model
{
    use HasFactory;

    public $guarded=['id,created_at,updated_at'];

    public function authors() {
        return $this->hasMany(Author::class)->withTimestamps();
    }

    public function genre() {
        return $this->belongsTo(Genre::class);
    }
}
```

3.4.4 Contact

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Contact extends Model
{
    use HasFactory;

    public $guarded=['id','created_at','updated_at'];
}
```

3.4.5 Genre

```
<?php
```

```

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Genre extends Model
{
    use HasFactory;

    public $guarded = ['id', 'created_at', 'updated_at'];

    public function books() {
        return $this->hasMany(Book::class);
    }
}

```

3.4.6 Role

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Role extends Model
{
    use HasFactory;
}

```

3.4.7 User

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class User extends Model
{
    use HasFactory;
    public $guarded=['id','created_at','updated_at'];
    protected $hidden=['password'];

    public function role() {
        return $this->belongsTo(Role::class, "role_id", "id");
    }
}

```

3.5 Migrations

3.5.1 Roles table

```

<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateRolesTable extends Migration
{

```

```

/**
 * Run the migrations.
 *
 * @return void
 */
public function up()
{
    Schema::create('roles', function (Blueprint $table) {
        $table->id();
        $table->string('role_name');
        $table->timestamps();
    });
}

/**
 * Reverse the migrations.
 *
 * @return void
 */
public function down()
{
    Schema::dropIfExists('roles');
}
}

```

3.5.2 Users table

```

<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateUsersTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('users', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('email')->unique();
            $table->string('password');
            $table->foreignId('role_id')->default(1)->references('id')->on('roles');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('users');
    }
}

```

3.5.3 Genres table

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateGenresTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('genres', function (Blueprint $table) {
            $table->id();
            $table->string('genre_name')->unique();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('genres');
    }
}
```

3.5.4 Authors table

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateAuthorsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('authors', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
    }
```

```

{
    Schema::dropIfExists('authors');
}
}

```

3.5.5 Books table

```

<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateBooksTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('books', function (Blueprint $table) {
            $table->id();
            $table->string('book_name');
            $table->integer('price');
            $table->text('description');
            $table->string('image');
            $table->string('publish_year');
            $table->foreignId('genre_id')->references('id')->on('genres');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('books');
    }
}

```

3.5.6 Author_book table

```

<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateAuthorBookTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('author_book', function (Blueprint $table) {
            $table->id();

```

```

        $table->foreignId('author_id')->references('id')->on('authors');
        $table->foreignId('book_id')->references('id')->on('books');
        $table->timestamps();
    });
}

/**
 * Reverse the migrations.
 *
 * @return void
 */
public function down()
{
    Schema::dropIfExists('author_book');
}
}

```

3.5.7 Contacts table

```

<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateContactsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('contacts', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('email');
            $table->string('message',200);
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('contacts');
    }
}

```

3.5.8 Activities table

```

<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateActivitiesTable extends Migration
{

```

```

/**
 * Run the migrations.
 *
 * @return void
 */
public function up()
{
    Schema::create('activities', function (Blueprint $table) {
        $table->id();
        $table->string('type');
        $table->string('user_name');
        $table->timestamps();
    });
}

/**
 * Reverse the migrations.
 *
 * @return void
 */
public function down()
{
    Schema::dropIfExists('activities');
}
}

```

3.6. Seeders

3.6.1 BookSeeder

```

<?php

namespace Database\Seeders;

use App\Models\Author;
use App\Models\Book;
use Illuminate\Database\Seeder;

class BookSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    public function run()
    {
        $book = Book::create([
            'book_name'=>'The Haunting of Hill House',
            'price'=>14,
            'description'=>'First published in 1959, Shirley Jackson\'s The Haunting of Hill House has been hailed as a perfect work of unnerving terror. It is the story of four seekers who arrive at a notoriously unfriendly pile called Hill House: Dr. Montague, an occult scholar looking for solid evidence of a "haunting"; Theodora, his lighthearted assistant; Eleanor, a friendless, fragile young woman well acquainted with poltergeists; and Luke, the future heir of Hill House. At first, their stay seems destined to be merely a spooky encounter with inexplicable phenomena. But Hill House is gathering its powers—and soon it will choose one of them to make its own.',
            'image'=>'the_haunting_of_hill_house.jpg',
            'publish_year'=>'2006',
            'genre_id'=>1
        ]);

        $author = Author::firstOrCreate(['name'=>'Shirley Jackson']);
        $book->authors()->sync($author->id);
    }
}

```

```

$book = Book::create([
    'book_name'=>"Gwendy's Final Task",
    'price'=>9,
    'description'=>'When Gwendy Peterson was twelve, a mysterious stranger
named Richard Farris gave her a mysterious box for safekeeping. It offered treats and
vintage coins, but it was dangerous. Pushing any of its seven colored buttons promised
death and destruction. Years later, the button box entered Gwendy\'s life again. A
successful novelist and a rising political star, she was once again forced to deal with
the temptation that box represented. Now, evil forces seek to possess the button box
and it is up to Senator Gwendy Peterson to keep it from them. At all costs. But where
can you hide something from such powerful entities? In Gwendy\'s Final Task, "horror
giants" Stephen King and Richard Chizmar take us on a journey from Castle Rock to
another famous cursed Maine city to the MF-1 space station, where Gwendy must execute a
secret mission to save the world. And, maybe, all worlds.',
    'image'=>"gwendy's_final_task.jpg",
    'publish_year'=>'2022',
    'genre_id'=>1
]);

$author = Author::firstOrCreate(['name'=>'Stephen King']);
$author2 = Author::firstOrCreate(['name'=>'Richard Chizmar']);
$book->authors()->sync([$author->id,$author2->id]);

$book = Book::create([
    'book_name'=>'Frankenstein',
    'price'=>15,
    'description'=>'Stepping far afield from his medical studies, Victor
Frankenstein brings to life a human form he has fashioned from scavenged body parts.
Horrorified by his achievement, he turns his back on his creation, only to learn the
danger of such neglect. Written when Mary Shelley was only 20 years old, Frankenstein
has been hailed as both a landmark of Gothic horror fiction and the first modern
science fiction story.',
    'image'=>'frankenstein.jpg',
    'publish_year'=>'2015',
    'genre_id'=>1
]);

$author = Author::firstOrCreate(['name'=>'Mary Shelley']);
$book->authors()->sync($author->id);

$book = Book::create([
    'book_name'=>'The Picture of Dorian Gray',
    'price'=>15,
    'description'=>'Horror hides behind an attractive face in The Picture of
Dorian Gray, Oscar Wilde\'s tale of a notorious Victorian libertine and his life of
evil excesses. Though Dorian\'s hedonistic indulgences leave no blemish on his ageless
features, the painted portrait imbued with his soul proves a living catalogue of
corruption, revealing in its every new line and lesion the manifold sins he has
committed. Desperate to hide the physical evidence of his unregenerate spirit, Dorian
will stop at nothing-not even murder-to keep his picture\'s existence a secret.',
    'image'=>'the_picture_of_dorian_gray.jpg',
    'publish_year'=>'2015',
    'genre_id'=>1
]);

$author = Author::firstOrCreate(['name'=>'Oscar Wilde']);
$book->authors()->sync($author->id);

$book = Book::create([
    'book_name'=>'The Only Good Indians',
    'price'=>13,

```



```

        'description'=>'At the crossroads of horror, poetry and history sits The
Only Good Indians. These three paths might seem inconceivable considering the first
road mentioned. No talk here of what will keep you up at night. "Poetic" should be
called out in Jones' writing. He gives distinct voice to his four main characters. You
will forever hear their stories in your mind, long after turning the last page. And in
those stories is a deep history of Native American culture. Sometimes the true horror
in a story goes beyond the passages that keep you up at night. First rate all around
for this novel.',
        'image'=>'the_only_good_indians.jpg',
        'publish_year'=>'2021',
        'genre_id'=>1
    ]);

    $author = Author::firstOrCreate(['name'=>'Stephen Graham Jones']);
    $book->authors()->sync($author->id);

    $book = Book::create([
        'book_name'=>'The Shining',
        'price'=>16,
        'description'=>'Jack Torrance's new job at the Overlook Hotel is the
perfect chance for a fresh start. As the off-season caretaker at the atmospheric old
hotel, he'll have plenty of time to spend reconnecting with his family and working on
his writing. But as the harsh winter weather sets in, the idyllic location feels ever
more remote . . . and more sinister. And the only one to notice the strange and
terrible forces gathering around the Overlook is Danny Torrance, a uniquely gifted
five-year-old.',
        'image'=>'the_shining.jpg',
        'publish_year'=>'2013',
        'genre_id'=>1
    ]);

    $author = Author::firstOrCreate(['name'=>'Stephen King']);
    $book->authors()->sync($author->id);

    $book = Book::create([
        'book_name'=>'Edgar Allan Poe: Classic Stories',
        'price'=>11,
        'description'=>'Edgar Allan Poe was a master of the tale of psychological
horror and the author of what is considered the first modern detective story. This
anthology gathers more than 20 of Poe's groundbreaking tales of the macabre, among them
"The Tell-Tale Heart," "The Masque of the Red Death," and "The Fall of the House of
Usher." It also includes his trilogy of stories featuring detective C. Auguste Dupin:
"The Murders in the Rue Morgue," "The Mystery of Marie Rôget," and "The Purloined
Letter."',
        'image'=>'edgar_allan_poe_classic_stories.jpg',
        'publish_year'=>'2018',
        'genre_id'=>1
    ]);

    $author = Author::firstOrCreate(['name'=>'Edgar Allan Poe']);
    $book->authors()->sync($author->id);

    $book = Book::create([
        'book_name'=>'How the Grinch Stole Christmas!',
        'price'=>17,
        'description'=>'Grow your heart three sizes and get in on all of the
Grinch-mas cheer with this Christmas classic-the ultimate Dr. Seuss holiday book that
no collection is complete without! Every Who down in Who-ville liked Christmas a lot .
. . but the Grinch, who lived just north of Who-ville, did NOT! Not since "'Twas the
night before Christmas" has the beginning of a Christmas tale been so instantly
recognizable. This heartwarming story about the effects of the Christmas spirit will
grow even the coldest and smallest of hearts. Like mistletoe, candy canes, and
caroling, the Grinch is a mainstay of the holidays, and his story is the perfect gift

```

```

for readers young and old.',
  'image'=>'how_the_grinch_stole_christmas!.jpg',
  'publish_year'=>'2021',
  'genre_id'=>11
]);

$author = Author::firstOrCreate(['name'=>'Dr. Seuss']);
$book->authors()->sync($author->id);

$book = Book::create([
  'book_name'=>'The Lorax',
  'price'=>8,
  'description'=>'I am the Lorax. I speak for the trees. Dr. Seuss’s beloved
story teaches kids to treat the planet with kindness and stand up and speak up for
others. Experience the beauty of the Truffula Trees and the danger of taking our earth
for granted in a story that is timely, playful, and hopeful. The book’s final pages
teach us that just one small seed, or one small child, can make a difference. Printed
on recycled paper, this book is the perfect gift for Earth Day and for any child-or
child at heart—who is interested in recycling, advocacy, and the environment, or just
loves nature and playing outside. Unless someone like you cares a whole awful lot,
nothing is going to get better. It’s not.',
  'image'=>'the_lorax.jpg',
  'publish_year'=>'1971',
  'genre_id'=>11
]);

$author = Author::firstOrCreate(['name'=>'Dr. Seuss']);
$book->authors()->sync($author->id);

$book = Book::create([
  'book_name'=>'Barnyard Dance!',
  'price'=>7,
  'description'=>'Everybody sing along—because it’s time to do-si-do in the
barnyard with a high-spirited animal crew! From Boynton on Board, the bestselling
series of board books, here is BARNYARD DANCE, with Sandra Boynton’s twirling pigs,
fiddle-playing cows, and other unforgettable animals. Extra-big, extra-fat, and extra-
fun, BARNYARD DANCE features lively rhyming text and a die-cut cover that reveals the
wacky characters inside. Guaranteed to get kids and adults stomping their feet.',
  'image'=>'barnyard_dance!.jpg',
  'publish_year'=>'1993',
  'genre_id'=>11
]);

$author = Author::firstOrCreate(['name'=>'Sandra Boynton']);
$book->authors()->sync($author->id);

$book = Book::create([
  'book_name'=>'Wingbearer',
  'price'=>9,
  'description'=>'Zuli is extraordinary—she just doesn’t realize it yet.
Raised by mystical bird spirits in the branches of the Great Tree, she’s never ventured
beyond this safe haven. She’s never had to. Until now. When a sinister force threatens
the life-giving magic of the tree, Zuli, along with her guardian owl, Frowly, must get
to the root of it. So begins an adventure bigger than anything Zuli could’ve ever
imagined—one that will bring her, along with some newfound friends, face-to-face with
an ancient dragon, the so-called Witch-Queen, and most surprisingly of all: her true
identity.',
  'image'=>'wingbearer.jpg',
  'publish_year'=>'2022',
  'genre_id'=>11
]);

$author = Author::firstOrCreate(['name'=>'Marjorie Liu']);
$book->authors()->sync($author->id);

```

```

$book = Book::create([
    'book_name'=>'The Tale of Peter Rabbit',
    'price'=>5,
    'description'=>'Once upon a time there were four little Rabbits, and their
names were – Flopsy, Mopsy, Cotton-tail, and Peter. The text and illustrations of the
classic Tale of Peter Rabbit, loved by children around the world for generations,
appear in full. Large, easy-to-read text makes this perfect for storytime and bedtime,
and the classic story is sure to be a favorite.',
    'image'=>'the_tale_of_peter_rabbit.jpg',
    'publish_year'=>'2004',
    'genre_id'=>11
]);

$author = Author::firstOrCreate(['name'=>'Beatrix Potter']);
$book->authors()->sync($author->id);

$book = Book::create([
    'book_name'=>'Happy Easter, Biscuit!',
    'price'=>6,
    'description'=>'Hippity, hoppity...woof! Join Biscuit on an Easter
adventure, with flaps for your little egg hunters to lift! Biscuit\'s first Easter egg
hunt is going to be egg-stra special. Where are all the eggs hidden? Your little pup
will enjoy lifting the flaps to find the eggs! Happy Easter, Biscuit! is a fun Easter
basket and springtime gift. Biscuit and the little girl have been warmly embraced by
families for 25 years. Author Alyssa Satin Capucilli reflects: "It is my childhood
dream of having a dog, perhaps a somewhat universal dream, that fuels the emotional
life of the young child that cares so lovingly for Biscuit. And knowing that the fluid
imagination of a child believes a beloved pet can understand every word spoken to them,
the words \'woof, woof\' not only become the voice of a small yellow puppy, but of
every reader\'s invention as well."',
    'image'=>'happy_easter_biscuit!.jpg',
    'publish_year'=>'2020',
    'genre_id'=>11
]);

$author = Author::firstOrCreate(['name'=>'Alyssa Satin Capucilli']);
$book->authors()->sync($author->id);

$book = Book::create([
    'book_name'=>'Pieces of Her',
    'price'=>10,
    'description'=>'What if the person you thought you knew best turns out to
be someone you never knew at all . . . ? Andrea Cooper knows everything about her
mother Laura. She\'s knows she\'s spent her whole life in the small beachside town of
Gullaway Island; she knows she\'s never wanted anything more than to live a quiet life
as a pillar of the community; she knows she\'s never kept a secret in her life. Because
we all know our mothers, don\'t we? But all that changes when a Saturday afternoon trip
to the mall explodes into violence and Andrea suddenly sees a completely different side
to Laura. Because it turns out that before Laura was Laura, she was someone completely
different. For nearly thirty years she\'s been hiding from her previous identity, lying
low in the hope that no one will ever find her. But now she\'s been exposed, and nothing
will ever be the same again. Twenty-four hours later Laura is in the hospital, shot by
an intruder who\'s spent thirty years trying to track her down and discover what she
knows. Andrea is on a desperate journey following the breadcrumbs of her mother\'s past.
And if she can\'t uncover the secrets hidden there, there may be no future for either
one of them. . . .',
    'image'=>'pieces_of_her.jpg',
    'publish_year'=>'2018',
    'genre_id'=>21
]);

$author = Author::firstOrCreate(['name'=>'Karin Slaughter']);
$book->authors()->sync($author->id);

$book = Book::create([

```

```

        'book_name'=>'A Reference to Murder',
        'price'=>8,
        'description'=>'Charli Rae Warren is back home in Hazel Rock, Texas,
spending her time reading, collecting, and selling books—at least, the ones that don't
get eaten first by her father's pet armadillo. Running the family bookstore is a
demanding job, but solving murders on the side can be flat out dangerous . . . The Book
Barn is more than just a shop, it's a part of the community—and Charli is keeping busy
with a fundraising auction and the big rodeo event that's come to town. That includes
dealing with the Texas-sized egos of some celebrity cowboys, including Dalton Hibbs, a
blond, blue-eyed bull rider who gets overly rowdy one night with the local hairdresser
. . . and soon afterward, disappears into thin air. Dalton's brother also vanished
seven years ago—and Charli is thrown about whether Dalton is a villain or a victim.
After a close call with an assailant wielding a branding iron (that plays havoc with
her hair), and some strange vandalism on her property, she's going to have to team up
with the sheriff to untangle this mystery, before she gets gored . . .',
        'image'=>'a_reference_to_murder.jpg',
        'publish_year'=>'2017',
        'genre_id'=>21
    ]);

    $author = Author::firstOrCreate(['name'=>'Kym Roberts']);
    $book->authors()->sync($author->id);

    $book = Book::create([
        'book_name'=>'And Then There Were None',
        'price'=>3,
        'description'=>'Ten people, each with something to hide and something to
fear, are invited to a isolated mansion on Indian Island by a host who, surprisingly,
fails to appear. On the island they are cut off from everything but each other and the
inescapable shadows of their own past lives. One by one, the guests share the darkest
secrets of their wicked pasts. And one by one, they die... Which among them is the killer
and will any of them survive?',
        'image'=>'and_then_there_were_none.jpg',
        'publish_year'=>'2009',
        'genre_id'=>21
    ]);

    $author = Author::firstOrCreate(['name'=>'Agatha Cristie']);
    $book->authors()->sync($author->id);

    $book = Book::create([
        'book_name'=>'Cain's Jawbone',
        'price'=>8,
        'description'=>'Six murders. One hundred pages. Millions of possible
combinations... but only one is correct. Can you solve Torquemada's murder mystery? In
1934, the Observer's cryptic crossword compiler, Edward Powys Mathers (aka Torquemada),
released a novel that was simultaneously a murder mystery and the most fiendishly
difficult literary puzzle ever written. The pages have been printed in an entirely
haphazard order, but it is possible - through logic and intelligent reading - to sort
the pages into the only correct order, revealing six murder victims and their
respective murderers. Only three puzzlers have ever solved the mystery of Cain's
Jawbone: do you have what it takes to join their ranks? Please note: this puzzle is
extremely difficult and not for the faint-hearted.',
        'image'=>'cain's_jawbone.jpg',
        'publish_year'=>'2021',
        'genre_id'=>21
    ]);

    $author = Author::firstOrCreate(['name'=>'Edward Powys Mathers']);
    $book->authors()->sync($author->id);

    $book = Book::create([
        'book_name'=>'City Problems',
        'price'=>15,
        'description'=>'A moment of violence—a snap judgement—a life changed to the

```

core. Ed Runyon bolted from the NYPD after a runaway teen case fell through the cracks and turned into a nightmarish murder. Now, he's learned to bury the rage that consumed him, cope with depression, and enjoy life as a Mifflin County sheriff's detective in rural Ohio. Ed is trying to relax on his day off when Columbus PD Detective Shelly Beckworth comes to Mifflin County in search of a girl who vanished after a pop-up party. The clues are scarce—a few license plates, a phone shattered on the roadside—but the trail leads to Ed's neck of the woods. He tries to shove everything else aside to keep this case from ending in another tragedy, but a cop can't pick and choose which calls to duty he'll answer. Frustrated, Ed watches a happy ending slip beyond sight—this one he cannot run away from. Charging forward, Ed breaks rules and takes risks leading to a bloody confrontation where everything he believes as a cop and every ghost in his head clash—a moment of avenging violence that will ultimately change his life to the core.',

```
'image'=>'city_problems.jpg',
'publish_year'=>'2021',
'genre_id'=>21
]);
```

```
$author = Author::firstOrCreate(['name'=>'Steve Goble']);
$book->authors()->sync($author->id);
```

```
$book = Book::create([
    'book_name'=>'The Maid',
    'price'=>13,
    'description'=>'Molly Gray is not like everyone else. She struggles with
social skills and misreads the intentions of others. Her gran used to interpret the
world for her, codifying it into simple rules that Molly could live by. Since Gran died
a few months ago, twenty-five-year-old Molly has been navigating life's complexities
all by herself. No matter—she throws herself with gusto into her work as a hotel maid.
Her unique character, along with her obsessive love of cleaning and proper etiquette,
make her an ideal fit for the job. She delights in donning her crisp uniform each
morning, stocking her cart with miniature soaps and bottles, and returning guest rooms
at the Regency Grand Hotel to a state of perfection. But Molly's orderly life is
upended the day she enters the suite of the infamous and wealthy Charles Black, only to
find it in a state of disarray and Mr. Black himself dead in his bed. Before she knows
what's happening, Molly's unusual demeanor has the police targeting her as their lead
suspect. She quickly finds herself caught in a web of deception, one she has no idea
how to untangle. Fortunately for Molly, friends she never knew she had unite with her
in a search for clues to what really happened to Mr. Black—but will they be able to
find the real killer before it's too late? A Clue-like, locked-room mystery and a
heartwarming journey of the spirit, The Maid explores what it means to be the same as
everyone else and yet entirely different—and reveals that all mysteries can be solved
through connection to the human heart.',
```

```
'image'=>'the_maid.jpg',
'publish_year'=>'2022',
'genre_id'=>21
]);
```

```
$author = Author::firstOrCreate(['name'=>'Nita Prose']);
$book->authors()->sync($author->id);
```

```
$book = Book::create([
    'book_name'=>'The Illiad & The Odyssey',
    'price'=>20,
    'description'=>'A classic keepsake for fans of Greek mythology, as well as
all great literature, this collection is the perfect addition to any library. The
influence of these two works can be seen far and wide, from James Joyce's Ulysses to
the movie sensation Troy, starring Brad Pitt.',
    'image'=>'the_illiad & _the_odyssey.jpg',
    'publish_year'=>'2016',
    'genre_id'=>31
]);
```

```
$author = Author::firstOrCreate(['name'=>'Homer']);
$book->authors()->sync($author->id);
```

```

$book = Book::create([
    'book_name'=>'The Rose That Grew From Concrete',
    'price'=>14,
    'description'=>'Tupac Shakur\'s most intimate and honest thoughts were
uncovered only after his death with the instant classic The Rose That Grew from
Concrete. His talent was unbounded a raw force that commanded attention and respect.
His death was tragic—a violent homage to the power of his voice. His legacy is
indomitable—as vibrant and alive today as it has ever been. For the first time in
paperback, this collection of deeply personal poetry is a mirror into the legendary
artist\'s enigmatic world and its many contradictions. Written in his own hand from the
time he was nineteen, these seventy-two poems embrace his spirit, his energy—and his
ultimate message of hope.',
    'image'=>'the_rose_that_grew_from_concrete.jpg',
    'publish_year'=>'2009',
    'genre_id'=>31
]);

$author = Author::firstOrCreate(['name'=>'Tupac Shakur']);
$book->authors()->sync($author->id);

$book = Book::create([
    'book_name'=>'Call Us What We Carry',
    'price'=>21,
    'description'=>'Formerly titled The Hill We Climb and Other Poems, the
luminous poetry collection by #1 New York Times bestselling author and presidential
inaugural poet Amanda Gorman captures a shipwrecked moment in time and transforms it
into a lyric of hope and healing. In Call Us What We Carry, Gorman explores history,
language, identity, and erasure through an imaginative and intimate collage. Harnessing
the collective grief of a global pandemic, this beautifully designed volume features
poems in many inventive styles and structures and shines a light on a moment of
reckoning. Call Us What We Carry reveals that Gorman has become our messenger from the
past, our voice for the future.',
    'image'=>'call_us_what_we_carry.jpg',
    'publish_year'=>'2021',
    'genre_id'=>31
]);

$author = Author::firstOrCreate(['name'=>'Amanda Gorman']);
$book->authors()->sync($author->id);

$book = Book::create([
    'book_name'=>'Pillow Thoughts',
    'price'=>15,
    'description'=>'Make a cup of tea and let yourself feel. Pillow Thoughts is
a collection of poetry and prose about heartbreak, love, and raw emotions. It is
divided into sections to read when you feel you need them most.',
    'image'=>'pillow_thoughts.jpg',
    'publish_year'=>',
    'genre_id'=>31
]);

$author = Author::firstOrCreate(['name'=>'Courtney Peppernell']);
$book->authors()->sync($author->id);

$book = Book::create([
    'book_name'=>'Milk and Honey',
    'price'=>12,
    'description'=>'The book is divided into four chapters, and each chapter
serves a different purpose. Deals with a different pain. Heals a different heartache.
milk and honey takes readers through a journey of the most bitter moments in life and
finds sweetness in them because there is sweetness everywhere if you are just willing
to look.',
    'image'=>'milk_and_honey.jpg',
    'publish_year'=>'2015',

```

```

        'genre_id'=>31
    });

    $author = Author::firstOrCreate(['name'=>'Rupi Kaur']);
    $book->authors()->sync($author->id);

    $book = Book::create([
        'book_name'=>'Home Body',
        'price'=>16,
        'description'=>'There is no one who knows the power of the written and
spoken word better than Rupri Kaur. In Home Body, Kaur\'s poems are compact. Each one,
however, is full of the exact right words. Words that help. Words that heal. Kaur\'s
accompanying illustrations add to the sensitivity of the material. Line drawings as
delicate as the poems. Combined with the poems, they create a strength. By going so
deep into life\'s experience, Kaur shows us how to resurface the strength we always
knew we had. This book is one of the bravest works of poetry you will ever encounter.',
        'image'=>'home_body.jpg',
        'publish_year'=>'2020',
        'genre_id'=>31
    ]);

    $author = Author::firstOrCreate(['name'=>'Rupi Kaur']);
    $book->authors()->sync($author->id);

    $book = Book::create([
        'book_name'=>'What If It's Us',
        'price'=>11,
        'description'=>'Critically acclaimed and bestselling authors Becky
Albertalli and Adam Silvera combine their talents in this smart, funny, heartfelt
collaboration about two very different boys who can't decide if the universe is pushing
them together—or pulling them apart. ARTHUR is only in New York for the summer, but if
Broadway has taught him anything, it's that the universe can deliver a showstopping
romance when you least expect it. BEN thinks the universe needs to mind its business.
If the universe had his back, he wouldn't be on his way to the post office carrying a
box of his ex-boyfriend's things. But when Arthur and Ben meet-cute at the post office,
what exactly does the universe have in store for them . . . ?',
        'image'=>'what_if_it's_us.jpg',
        'publish_year'=>'2020',
        'genre_id'=>41
    ]);

    $author = Author::firstOrCreate(['name'=>'Becky Albertalli']);
    $author2 = Author::firstOrCreate(['name'=>'Adam Silvera']);
    $book->authors()->sync([$author->id,$author2->id]);

    $book = Book::create([
        'book_name'=>'The Love Hypothesis',
        'price'=>12,
        'description'=>'When a fake relationship between scientists meets the
irresistible force of attraction, it throws one woman\'s carefully calculated theories
on love into chaos. As a third-year Ph.D. candidate, Olive Smith doesn\'t believe in
lasting romantic relationships—but her best friend does, and that\'s what got her into
this situation. Convinced Anh that Olive is dating and well on her way to a happily
ever after was always going to take more than hand-wavy Jedi mind tricks: Scientists
require proof. So, like any self-respecting biologist, Olive panics and kisses the
first man she sees. That man is none other than Adam Carlsen, a young hotshot
professor—and well-known ass. Which is why Olive is positively floored when Stanford\'s
reigning lab tyrant agrees to keep her charade a secret and be her fake boyfriend. But
when a big science conference goes haywire, putting Olive\'s career on the Bunsen
burner, Adam surprises her again with his unyielding support and even more
unyielding...six-pack abs. Suddenly their little experiment feels dangerously close to
combustion. And Olive discovers that the only thing more complicated than a hypothesis
on love is putting her own heart under the microscope.',
        'image'=>'the_love_hypothesis.jpg',
        'publish_year'=>'2021',
    ]);

```

```

        'genre_id'=>41
    });

    $author = Author::firstOrCreate(['name'=>'Ali Hazelwood']);
    $book->authors()->sync($author->id);

    $book = Book::create([
        'book_name'=>'Hook, Line, and Sinker',
        'price'=>12,
        'description'=>'King crab fisherman Fox Thornton has a reputation as a
sexy, carefree flirt. Everyone knows he's a guaranteed good time-in bed and out-and
that's exactly how he prefers it. Until he meets Hannah Bellinger. She's immune to his
charm and looks, but she seems to enjoy his... personality? And wants to be friends?
Bizarre. But he likes her too much to risk a fling, so platonic pals it is. Now,
Hannah\'s in town for work, crashing in Fox\'s spare bedroom. She knows he\'s a
notorious ladies\' man, but they\'re definitely just friends. In fact, she\'s nursing a
hopeless crush on a colleague and Fox is just the person to help with her lackluster
love life. Armed with a few tips from Westport\'s resident Casanova, Hannah sets out to
catch her coworker\'s eye... yet the more time she spends with Fox, the more she wants him
instead. As the line between friendship and flirtation begins to blur, Hannah can\'t
deny she loves everything about Fox, but she refuses to be another notch on his
bedpost. Living with his best friend should have been easy. Except now she\'s walking
around in a towel, sleeping right across the hall, and Fox is fantasizing about waking
up next to her for the rest of his life and... and... man overboard! He\'s fallen for her,
hook, line, and sinker. Helping her flirt with another guy is pure torture, but maybe
if Fox can tackle his inner demons and show Hannah he\'s all in, she\'ll choose him
instead?',
        'image'=>'hook_line_and_sinker.jpg',
        'publish_year'=>'2022',
        'genre_id'=>41
    ]);

    $author = Author::firstOrCreate(['name'=>'Tessa Bailey']);
    $book->authors()->sync($author->id);

    $book = Book::create([
        'book_name'=>'A Will and a Way',
        'price'=>7,
        'description'=>'When her beloved Uncle Jolley died, Pandora McVie couldn't
imagine her life without him-only to discover that he planned for her future by leaving
her $150 million. But to collect her inheritance, Pandora must spend six months in her
uncle's isolated Catskills mansion with her co-beneficiary, Michael Donohue. If being
set up on a half-year date that lasts through Christmas by a last will and testament
isn't humiliating enough, Pandora finds living with Michael intolerable-even as she
falls in love with him...',
        'image'=>'a_will_and_a_way.jpg',
        'publish_year'=>'2022',
        'genre_id'=>41
    ]);

    $author = Author::firstOrCreate(['name'=>'Nora Roberts']);
    $book->authors()->sync($author->id);

    $book = Book::create([
        'book_name'=>'A Touch of Darkness',
        'price'=>17,
        'description'=>'First book of Hadex x Persephone series. Persephone is the
Goddess of Spring by title only. The truth is, since she was a little girl, flowers
have shriveled at her touch. After moving to New Athens, she hopes to lead an
unassuming life disguised as a mortal journalist. Hades, God of the Dead, has built a
gambling empire in the mortal world and his favorite bets are rumored to be impossible.
After a chance encounter with Hades, Persephone finds herself in a contract with the
God of the Dead and the terms are impossible: Persephone must create life in the
Underworld or lose her freedom forever. The bet does more than expose Persephone\'s
failure as a goddess, however. As she struggles to sow the seeds of her freedom, love

```



```

for the God of the Dead grows-and it\'s forbidden.',
  'image'=>'a_touch_of_darkness.jpg',
  'publish_year'=>'2021',
  'genre_id'=>41
]);

$author = Author::firstOrCreate(['name'=>'Scarlett St. Clair']);
$book->authors()->sync($author->id);

$book = Book::create([
  'book_name'=>'A Touch of Ruin',
  'price'=>13,
  'description'=>'Second book of Hades x Persefone series. Persephone\'s
relationship with Hades has gone public and the resulting media storm disrupts her
normal life and threatens to expose her as the Goddess of Spring. Hades, God of the
Dead, is burdened by a hellish past that everyone\'s eager to expose in an effort to
warn Persephone away. Things only get worse when a horrible tragedy leaves
Persephone\'s heart in ruin and Hades refusing to help. Desperate, she takes matters
into her own hands, striking bargains with severe consequences. Faced with a side of
Hades she never knew and crushing loss, Persephone wonders if she can truly become
Hades\' queen.',
  'image'=>'a_touch_of_ruin.jpg',
  'publish_year'=>'2021',
  'genre_id'=>41
]);

$author = Author::firstOrCreate(['name'=>'Scarlett St. Clair']);
$book->authors()->sync($author->id);

$book = Book::create([
  'book_name'=>'A Touch of Malice',
  'price'=>15,
  'description'=>'Third book of Hades x Persefone series. Persephone and
Hades are engaged. In retaliation, Demeter summons a snowstorm that cripples New
Greece, and refuses to lift the blizzard unless her daughter calls off her engagement.
When the Olympians intervene, Persephone finds her future in the hands of ancient gods,
and they are divided. Do they allow Persephone to marry Hades and go to war with
Demeter or prohibit their union and take up arms against the God of the Dead? Nothing
is certain but the promise of war.',
  'image'=>'a_touch_of_malice.jpg',
  'publish_year'=>'2021',
  'genre_id'=>41
]);

$author = Author::firstOrCreate(['name'=>'Scarlett St. Clair']);
$book->authors()->sync($author->id);

$book = Book::create([
  'book_name'=>'Dune',
  'price'=>9,
  'description'=>'Set on the desert planet Arrakis, Dune is the story of the
boy Paul Atreides, heir to a noble family tasked with ruling an inhospitable world
where the only thing of value is the "spice" melange, a drug capable of extending life
and enhancing consciousness. Coveted across the known universe, melange is a prize
worth killing for.... When House Atreides is betrayed, the destruction of Paul\'s family
will set the boy on a journey toward a destiny greater than he could ever have
imagined. And as he evolves into the mysterious man known as Muad\'Dib, he will bring to
fruition humankind\'s most ancient and unattainable dream. A stunning blend of adventure
and mysticism, environmentalism and politics, Dune won the first Nebula Award, shared
the Hugo Award, and formed the basis of what is undoubtedly the grandest epic in
science fiction.',
  'image'=>'dune.jpg',
  'publish_year'=>'2003',
  'genre_id'=>51
]);

```

```

$author = Author::firstOrCreate(['name'=>'Frank Herbert']);
$book->authors()->sync($author->id);

$book = Book::create([
    'book_name'=>"Harry Potter and the Sorcerer's Stone",
    'price'=>11,
    'description'=>'Harry Potter spent ten long years living with Mr. and Mrs.
Dursley, an aunt and uncle whose outrageous favoritism of their perfectly awful son
Dudley leads to some of the most inspired dark comedy since Charlie and the Chocolate
Factory. But fortunately for Harry, he\'s about to be granted a scholarship to a unique
boarding school called THE HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY, where he will
become a school hero at the game of Quidditch (a kind of aerial soccer played high
above the ground on broomsticks), he will make some wonderful friends, and,
unfortunately, a few terrible enemies. For although he seems to be getting your run-of-
the-mill boarding school experience (well, ok, even that\'s pretty darn out of the
ordinary), Harry Potter has a destiny that he was born to fulfill. A destiny that
others would kill to keep him from.',
    'image'=>"harry_potter_and_the_sorcerer's_stone.jpg",
    'publish_year'=>'1998',
    'genre_id'=>51
]);

$author = Author::firstOrCreate(['name'=>'J. K. Rowling']);
$book->authors()->sync($author->id);

$book = Book::create([
    'book_name'=>'Harry Potter and the Chamber of Secrets',
    'price'=>11,
    'description'=>'The Dursleys were so mean and hideous that summer that all
Harry Potter wanted was to get back to the Hogwarts School for Witchcraft and Wizardry.
But just as he\'s packing his bags, Harry receives a warning from a strange, impish
creature named Dobby who says that if Harry Potter returns to Hogwarts, disaster will
strike. And strike it does. For in Harry\'s second year at Hogwarts, fresh torments and
horrors arise, including an outrageously stuck-up new professor, Gilderoy Lockhart, a
spirit named Moaning Myrtle who haunts the girls\' bathroom, and the unwanted
attentions of Ron Weasley\'s younger sister, Ginny. But each of these seem minor
annoyances when the real trouble begins, and someone - or something - starts turning
Hogwarts students to stone. Could it be Draco Malfoy, a more poisonous rival than ever?
Could it possibly be Hagrid, whose mysterious past is finally told? Or could it be the
one everyone at Hogwarts most suspects... Harry Potter himself!',
    'image'=>'harry_potter_and_the_chamber_of_secrets.jpg',
    'publish_year'=>'2000',
    'genre_id'=>51
]);

$author = Author::firstOrCreate(['name'=>'J. K. Rowling']);
$book->authors()->sync($author->id);

$book = Book::create([
    'book_name'=>'Harry Potter and the Prisoner of Azkaban',
    'price'=>11,
    'description'=>'For twelve long years, the dread fortress of Azkaban held
an infamous prisoner named Sirius Black. Convicted of killing thirteen people with a
single curse, he was said to be the heir apparent to the Dark Lord, Voldemort. Now he
has escaped, leaving only two clues as to where he might be headed: Harry Potter\'s
defeat of You-Know-Who was Black\'s downfall as well. And the Azkaban guards heard
Black muttering in his sleep, "He\'s at Hogwarts . . . he\'s at Hogwarts." Harry Potter
isn\'t safe, not even within the walls of his magical school, surrounded by his
friends. Because on top of it all, there may well be a traitor in their midst.',
    'image'=>'harry_potter_and_the_prisoner_of_azkaban.jpg',
    'publish_year'=>'2001',
    'genre_id'=>51
]);

```

```

$author = Author::firstOrCreate(['name'=>'J. K. Rowling']);
$book->authors()->sync($author->id);

$book = Book::create([
    'book_name'=>'Harry Potter and the Goblet of Fire',
    'price'=>13,
    'description'=>'Harry Potter is midway through his training as a wizard and his coming of age. Harry wants to get away from the pernicious Dursleys and go to the International Quidditch Cup. He wants to find out about the mysterious event that\'s supposed to take place at Hogwarts this year, an event involving two other rival schools of magic, and a competition that hasn\'t happened for a hundred years. He wants to be a normal, fourteen-year-old wizard. But unfortunately for Harry Potter, he\'s not normal - even by wizarding standards. And in his case, different can be deadly.',
    'image'=>'harry_potter_and_the_goblet_of_fire.jpg',
    'publish_year'=>'2002',
    'genre_id'=>51
]);

$author = Author::firstOrCreate(['name'=>'J. K. Rowling']);
$book->authors()->sync($author->id);

$book = Book::create([
    'book_name'=>'Harry Potter and the Order of the Phoenix',
    'price'=>13,
    'description'=>'The book that took the world by storm....In his fifth year at Hogwart\'s, Harry faces challenges at every turn, from the dark threat of He-Who-Must-Not-Be-Named and the unreliability of the government of the magical world to the rise of Ron Weasley as the keeper of the Gryffindor Quidditch Team. Along the way he learns about the strength of his friends, the fierceness of his enemies, and the meaning of sacrifice.',
    'image'=>'harry_potter_and_the_order_of_the_phoenix.jpg',
    'publish_year'=>'2004',
    'genre_id'=>51
]);

$author = Author::firstOrCreate(['name'=>'J. K. Rowling']);
$book->authors()->sync($author->id);

$book = Book::create([
    'book_name'=>'Harry Potter and the Half-Blood Prince',
    'price'=>13,
    'description'=>'Lord Voldemort has returned to power, and his wrath has been felt in both the Muggle and Wizarding worlds. Severus Snape, long considered an enemy of Voldemort and a member of Dumbledore\'s anti-Voldemort coalition, the Order of the Phoenix, meets with Narcissa Malfoy, mother of Draco and wife of Lucius, an imprisoned Death Eater. Snape makes an Unbreakable Vow to Narcissa, promising to protect her son, Draco. Dumbledore heads to 4 Privet Drive to collect Harry from his aunt and uncle. On their way to the Burrow, Harry and Dumbledore stop to recruit Horace Slughorn to return to teaching at Hogwarts. Harry is reunited with his best friends, Ron and Hermione. When shopping for schoolbooks, Harry runs into Draco and follows him to Borgin and Burkes, where he overhears Draco threatening Borgin and insisting that he fix an unknown object. Harry is instantly suspicious of Draco, whom he believes to be a Death Eater, just like his father. The students return to school, and Dumbledore announces that Snape will be teaching Defense Against the Dark Arts, much to Harry\'s surprise.',
    'image'=>'harry_potter_and_the_half-blood_prince.jpg',
    'publish_year'=>'2006',
    'genre_id'=>51
]);

$author = Author::firstOrCreate(['name'=>'J. K. Rowling']);
$book->authors()->sync($author->id);

$book = Book::create([
    'book_name'=>'Harry Potter and the Deathly Hallows',

```

```

        'price'=>15,
        'description'=>'At Malfoy Manor, Snape tells Voldemort the date that
Harry's friends are planning to move him from the house on Privet Drive to a new safe
location, so that Voldemort can capture Harry en route. As Harry packs to leave Privet
Drive, he reads two obituaries for Dumbledore, both of which make him think that he
didn't know Dumbledore as well as he should have. Downstairs, he bids good-bye to the
Dursleys for the final time, as the threat of Voldemort forces them to go into hiding
themselves. The Order of the Phoenix, led by Alastor "Mad-Eye" Moody, arrives to take
Harry to his new home at the Weasleys' house, the Burrow. Six of Harry's friends take
Polyjuice Potion to disguise themselves as Harry and act as decoys, and they all fly
off in different directions. The Death Eaters, alerted to their departure by Snape,
attack Harry and his friends. Voldemort chases Harry down, but Harry's wand fends
Voldemort off, seemingly without Harry's help.',
        'image'=>'harry_potter_and_the_deathly_hallows.jpg',
        'publish_year'=>'2009',
        'genre_id'=>51
    ]);

    $author = Author::firstOrCreate(['name'=>'J. K. Rowling']);
    $book->authors()->sync($author->id);

    $book = Book::create([
        'book_name'=>'The Fellowship of the Ring: The Lord of the Rings: Part One',
        'price'=>9,
        'description'=>'The opening novel of The Lord of the Rings—the greatest
fantasy epic of all time—which continues in The Two Towers and The Return of the King.
The dark, fearsome Ringwraiths are searching for a Hobbit. Frodo Baggins knows that
they are seeking him and the Ring he bears—the Ring of Power that will enable evil
Sauron to destroy all that is good in Middle-earth. Now it is up to Frodo and his
faithful servant, Sam, with a small band of companions, to carry the Ring to the one
place it can be destroyed: Mount Doom, in the very center of Sauron's realm.',
        'image'=>'the_fellowship_of_the_ring_the_lord_of_the_rings_part_one.jpg',
        'publish_year'=>'1986',
        'genre_id'=>51
    ]);

    $author = Author::firstOrCreate(['name'=>'J. R. R. Tolkien']);
    $book->authors()->sync($author->id);

    $book = Book::create([
        'book_name'=>'The Two Towers: The Lord of the Rings: Part Two',
        'price'=>9,
        'description'=>'The middle novel in The Lord of the Rings—the greatest
fantasy epic of all time—which began in The Fellowship of the Ring, and which reaches
its magnificent climax in The Return of the King. The Fellowship is scattered. Some
brace hopelessly for war against the ancient evil of Sauron. Others must contend with
the treachery of the wizard Saruman. Only Frodo and Sam are left to take the One Ring,
ruler of the accursed Rings of Power, to be destroyed in Mordor, the dark realm where
Sauron is supreme. Their guide is Gollum, deceitful and obsessive slave to the
corruption of the Ring.',
        'image'=>'the_two_towers_the_lord_of_the_rings_part_two.jpg',
        'publish_year'=>'1986',
        'genre_id'=>51
    ]);

    $author = Author::firstOrCreate(['name'=>'J. R. R. Tolkien']);
    $book->authors()->sync($author->id);

    $book = Book::create([
        'book_name'=>'The Return of The King: The Lord of the Rings: Part Three',
        'price'=>9,
        'description'=>'The awesome conclusion to The Lord of the Rings—the
greatest fantasy epic of all time—which began in The Fellowship of the Ring and The Two
Towers. While the evil might of the Dark Lord Sauron swarms out to conquer all Middle-
earth, Frodo and Sam struggle deep into Mordor, seat of Sauron's power. To defeat the

```

```

Dark Lord, the One Ring, ruler of the accursed Rings of Power, must be destroyed in the
fires of Mount Doom. But the way is impossibly hard, and Frodo is weakening. Weighed
down by the compulsion of the Ring, he begins finally to despair.',
  'image'=>'the_return_of_the_king_the_lord_of_the_rings_part_three.jpg',
  'publish_year'=>'1986',
  'genre_id'=>51
]);

$author = Author::firstOrCreate(['name'=>'J. R. R. Tolkien']);
$book->authors()->sync($author->id);

$book = Book::create([
  'book_name'=>'Verity',
  'price'=>17,
  'description'=>'Lowen Ashleigh is a struggling writer on the brink of
financial ruin when she accepts the job offer of a lifetime. Jeremy Crawford, husband
of bestselling author Verity Crawford, has hired Lowen to complete the remaining books
in a successful series his injured wife is unable to finish. Lowen arrives at the
Crawford home, ready to sort through years of Verity's notes and outlines, hoping to
find enough material to get her started. What Lowen doesn't expect to uncover in the
chaotic office is an unfinished autobiography Verity never intended for anyone to read.
Page after page of bone-chilling admissions, including Verity's recollection of the
night her family was forever altered. Lowen decides to keep the manuscript hidden from
Jeremy, knowing its contents could devastate the already grieving father. But as
Lowen's feelings for Jeremy begin to intensify, she recognizes all the ways she could
benefit if he were to read his wife's words. After all, no matter how devoted Jeremy is
to his injured wife, a truth this horrifying would make it impossible for him to
continue loving her.',
  'image'=>'verity.jpg',
  'publish_year'=>'2021',
  'genre_id'=>61
]);

$author = Author::firstOrCreate(['name'=>'Colleen Hoover']);
$book->authors()->sync($author->id);

$book = Book::create([
  'book_name'=>'The Overnight Guest',
  'price'=>18,
  'description'=>'True crime writer Wylie Lark doesn't mind being snowed in
at the isolated farmhouse where she's retreated to write her new book. A cozy fire,
complete silence. It would be perfect, if not for the fact that decades earlier, at
this very house, two people were murdered in cold blood and a girl disappeared without
a trace. As the storm worsens, Wylie finds herself trapped inside the house, haunted by
the secrets contained within its walls—haunted by secrets of her own. Then she
discovers a small child in the snow just outside. After bringing the child inside for
warmth and safety, she begins to search for answers. But soon it becomes clear that the
farmhouse isn't as isolated as she thought, and someone is willing to do anything to
find them.',
  'image'=>'the_overnight_guest.jpg',
  'publish_year'=>'2022',
  'genre_id'=>61
]);

$author = Author::firstOrCreate(['name'=>'Heather Gudenkauf']);
$book->authors()->sync($author->id);

$book = Book::create([
  'book_name'=>'Cry Wolf',
  'price'=>14,
  'description'=>'A DEAD WOLF. A DRUG DEAL GONE WRONG. A LETHAL FEMALE
ASSASSIN. The first book in a new series by Hans Rosenfeldt, creator of the TV series
The Bridge as well as Netflix's Emmy Award-winning Marcella. Hannah Wester, a
policewoman in the remote northern town of Haparanda, Sweden, finds herself on the
precipice of chaos. When human remains are found in the stomach of a dead wolf, Hannah

```

knows that this summer won't be like any other. The remains are linked to a bloody drug deal across the border in Finland. But how did the victim end up in the woods outside of Haparanda? And where have the drugs and money gone? Hannah and her colleagues leave no stone unturned. But time is scarce and they aren't the only ones looking. When the secretive and deadly Katja arrives, unexpected and brutal events start to pile up. In just a few days, life in Haparanda is turned upside down. Not least for Hannah, who is finally forced to confront her own past.'

```
    'image'=>'cry_wolf.jpg',  
    'publish_year'=>'2021',  
    'genre_id'=>61  
]);  
  
$author = Author::firstOrCreate(['name'=>'Hans Rosenfeldt']);  
$book->authors()->sync($author->id);
```

```
$book = Book::create([  
    'book_name'=>'The Golden Couple',  
    'price'=>15,  
    'description'=>'Wealthy Washington suburbanites Marissa and Matthew Bishop seem to have it all-until Marissa is unfaithful. Beneath their veneer of perfection is a relationship riven by work and a lack of intimacy. She wants to repair things for the sake of their eight-year-old son and because she loves her husband. Enter Avery Chambers. Avery is a therapist who lost her professional license. Still, it doesn't stop her from counseling those in crisis, though they have to adhere to her unorthodox methods. And the Bishops are desperate. When they glide through Avery's door and Marissa reveals her infidelity, all three are set on a collision course. Because the biggest secrets in the room are still hidden, and it's no longer simply a marriage that's in danger.'
```

```
    'image'=>'the_golden_couple.jpg',  
    'publish_year'=>'2022',  
    'genre_id'=>61  
]);
```

```
$author = Author::firstOrCreate(['name'=>'Greer Hendricks']);  
$author2 = Author::firstOrCreate(['name'=>'Sarah Pekkanen']);  
$book->authors()->sync([$author->id,$author2->id]);
```

```
$book = Book::create([  
    'book_name'=>'Hostage',  
    'price'=>18,  
    'description'=>'You can save hundreds of lives. Or the one that matters most... From New York Times bestselling author Clare Mackintosh comes a claustrophobic thriller set over 20 hours on-board the inaugural nonstop flight from London to Sydney. Mina is trying to focus on her job as a flight attendant, not the problems with her five-year-old daughter back home, or the fissures in her marriage. But the plane has barely taken off when Mina receives a chilling note from an anonymous passenger, someone intent on ensuring the plane never reaches its destination: "The following instructions will save your daughter's life..." Someone needs Mina's assistance and knows exactly how to make her comply. When one passenger is killed and then another, Mina knows she must act. But which lives does she save: Her passengers...or her own daughter and husband who are in grave distress back at home? It's twenty hours to landing. A lot can happen in twenty hours.'
```

```
    'image'=>'hostage.jpg',  
    'publish_year'=>'2022',  
    'genre_id'=>61  
]);
```

```
$author = Author::firstOrCreate(['name'=>'Clare Mackintosh']);  
$book->authors()->sync($author->id);
```

```
$book = Book::create([  
    'book_name'=>'Killer View',  
    'price'=>8,  
    'description'=>'Kendra Michaels, blind before gaining her sight via a revolutionary surgical procedure, offers her razor-sharp senses to assist her friend
```

```

Jessie Mercado in a baffling case. An army vet and former bodyguard for the rich and famous, Jessie has faced all kinds of danger but one thing the motorcycle-riding private investigator has never encountered before is an incarceration consultant. Preparing wealthy people to go to prison is big business. When Owen Blake of Mamertine Consulting hires Mercado to find his missing partner, their suspect list is filled with recently released white-collar criminals, a few drug kingpins, and a couple of murderers to keep things interesting. As witnesses turn up dead and car chases leave destruction in their wake, Jessie and Kendra learn just how far someone will go to keep the fate of one man hidden. But why? Together they must hunt down the lethal secrets of Blake's company, hell-bent on staying one step ahead of disaster.',
        'image'=>'killer_view.jpg',
        'publish_year'=>'2022',
        'genre_id'=>61
    ]);

    $author = Author::firstOrCreate(['name'=>'Roy Johansen']);
    $book->authors()->sync($author->id);
}
}

```

3.6.2 DatabaseSeeder

```

<?php

namespace Database\Seeders;

use Illuminate\Database\Seeder;

class DatabaseSeeder extends Seeder
{
    /**
     * Seed the application's database.
     *
     * @return void
     */
    public function run()
    {
        $this->call([
            RoleSeeder::class,
            UserSeeder::class,
            GenreSeeder::class,
            BookSeeder::class
        ]);
    }
}

```

3.6.3 GenreSeeder

```

<?php

namespace Database\Seeders;

use App\Models\Genre;
use Illuminate\Database\Seeder;

class GenreSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    public function run()
    {

```

```

        Genre::create(['genre_name'=>'Horror']);
        Genre::create(['genre_name'=>'Kids']);
        Genre::create(['genre_name'=>'Mystery']);
        Genre::create(['genre_name'=>'Poetry']);
        Genre::create(['genre_name'=>'Romance']);
        Genre::create(['genre_name'=>'Sci-Fi & Fantasy']);
        Genre::create(['genre_name'=>'Thriller']);
    }
}

```

3.6.4 RoleSeeder

```

<?php

namespace Database\Seeders;

use App\Models\Role;
use Illuminate\Database\Seeder;

class RoleSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    public function run()
    {
        Role::create(['role_name'=>'user']);
        Role::create(['role_name'=>'admin']);
    }
}

```

3.6.5 UserSeeder

```

<?php

namespace Database\Seeders;

use App\Models\User;
use Illuminate\Database\Seeder;

class UserSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    public function run()
    {
        User::create([
            'name' => 'Petar Petrovic',
            'email' => 'pera@gmail.com',
            'password' => md5('pera123')
        ]);
        User::create([
            'name' => 'Laza Lazic',
            'email' => 'laza@gmail.com',
            'password' => md5('laza123')
        ]);

        User::create([

```



```

        'name' => 'Marko Markovic',
        'email' => 'admin@gmail.com',
        'password' => md5('admin123'),
        'role_id' => 11
    });
}
}

```

3.7 Views

3.7.1 Fixed

3.7.1.1 footer.blade.php

```

<div class="back-to-top d-flex justify-content-center align-items-center rounded-circle
hidden"><i class="fas fa-angle-up"></i></div>
<footer>
    <div class="container-fluid p-md-3 bg-dark text-light">
        <div class="container">
            <div class="row text-center">
                <div class="col-12 col-md-4 mb-3 mt-2 m-md-0">
                    <h4>Follow us</h4>
                    <ul>
                        <li><a href="https://www.facebook.com" target="blank"><i
class="fab fa-facebook-f"></i> Facebook</a></li>
                        <li><a href="https://www.instagram.com" target="blank"><i
class="fab fa-instagram"></i> Instagram</a></li>
                        <li><a href="https://www.youtube.com" target="blank"><i
class="fab fa-youtube"></i> YouTube</a></li>
                    </ul>
                </div>
                <div class="col-12 col-md-4 mb-3 m-md-0">
                    <h4>Working hours</h4>
                    <p><i class="fas fa-clock"></i> Workdays: 08-20h</p>
                    <p><i class="fas fa-clock"></i> Weekends: 09-18h</p>
                    <p><i class="fas fa-map-marker-alt"></i> Knez Mihailova 1,
Belgrade</p>
                </div>
                <div class="col-12 col-md-4">
                    <h4>Links</h4>
                    <ul>
                        <li><a href="{{asset('dokumentacija.pdf')}}" target="blank"><i
class="fas fa-sitemap"></i> Documentation</a></li>
                        <li><a href="{{route('author')}}"><i class="far fa-file-
code"></i> Author</a></li>
                        <li>&copy; Djordje Solomun 16/19</li>
                    </ul>
                </div>
            </div>
        </div>
    </div>
</footer>

```

3.7.1.2 head.blade.php

```

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="keywords" content="@yield('keywords')">
    <meta name="description" content="@yield('description')">
    <meta name="author" content="Djordje Solomun">
    <link rel="shortcut icon" href="{{asset('img/favicon.ico')}}" type="image/x-icon">
    <meta name="csrf-token" content="{{ csrf_token() }}">
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.5/dist/css/bootstrap.min.css" integrity="sha512-

```

```

HK5fgLBL+xu6dm/Ii3z4xhLSUyZgTT9tuc/hSrtw6uzJOvgRr2a9jyxxTlely+B+xFamJKVSTbpM/CuL7qxO8w=
=" crossorigin="anonymous" />
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css"
integrity="sha384-TX8t27EcRE3e/ihU7zmQxVncDAY5uIKz4rEkgIXeMed4M0jlfIDPvg6uqKI2xXr2"
crossorigin="anonymous">
<link rel="stylesheet" href="{{asset('css/style.css')}}">
<link href="https://cdnjs.cloudflare.com/ajax/libs/toastr.js/latest/toastr.css"
rel="stylesheet">
<title>Bookstore - @yield('title')</title>
</head>

```

3.7.1.3 nav.blade.php

```

<nav class="navbar sticky-top navbar-expand-lg navbar-dark bg-dark">
  <a class="navbar-brand" href="{{route('home')}}">Bookstore</a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarTogglerDemo02" aria-controls="navbarTogglerDemo02" aria-expanded="false"
aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="navbarTogglerDemo02">
    <ul class="navbar-nav mr-auto mt-2 mt-lg-0 w-100">
      <li class="nav-item @if(request()->routeIs('home')) active @endif">
        <a class="nav-link" href="{{route('home')}}">Home</a>
      </li>
      <li class="nav-item @if(request()->routeIs('books.index')) active @endif">
        <a class="nav-link" href="{{route('books.index')}}">Shop</a>
      </li>
      <li class="nav-item @if(request()->routeIs('contact.index')) active
@endif">
        <a class="nav-link" href="{{route('contact.index')}}">Contact</a>
      </li>
      @if(session()->has('user'))
        @if(session()->get('user')->role->role_name == 'user')
          <li class="nav-item @if(request()->routeIs('cart')) active @endif">
            <a class="nav-link" href="{{route('cart')}}">Cart<i class="fas
fa-shopping-cart"></i></a>
          </li>
        @else
          <li class="nav-item">
            <a class="nav-link" href="{{route('admin.home')}}">Admin
panel</a>
          </li>
        @endif
      <li class="nav-item ml-lg-auto">
        <span class="text-white">{{session()->get('user')->name}}</span>
        <a class="auth-button" href="{{route('logout')}}">Logout</a>
      </li>
      @else
        <li class="nav-item px-2 ml-lg-auto">
          <a class="auth-button" href="{{route('login')}}">Login</a>
        </li>
        <li class="nav-item py-1 py-lg-0">
          <a class="auth-button" href="{{route('register')}}">Register</a>
        </li>
      @endif
    </ul>
  </div>
</nav>

```

3.7.1.4 nav_admin.blade.php

```
<nav class="navbar sticky-top navbar-expand-lg navbar-dark bg-dark">
    <a class="navbar-brand" href="{{route('home')}}">Bookstore</a>
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarTogglerDemo02" aria-controls="navbarTogglerDemo02" aria-expanded="false"
aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarTogglerDemo02">
        <ul class="navbar-nav mr-auto mt-2 mt-lg-0 w-100">
            <li class="nav-item @if(request()->routeIs('admin.home')) active
@endif">
                <a class="nav-link" href="{{route('admin.home')}}">Admin home</a>
            </li>
            <li class="nav-item @if(request()->routeIs('admin.products')) active
@endif">
                <a class="nav-link" href="{{route('admin.products')}}">Admin
products</a>
            </li>
            <li class="nav-item @if(request()->routeIs('admin.genres')) active
@endif">
                <a class="nav-link" href="{{route('admin.genres')}}">Admin
genres</a>
            </li>
            <li class="nav-item @if(request()->routeIs('admin.users')) active
@endif">
                <a class="nav-link" href="{{route('admin.users')}}">Admin users</a>
            </li>
            <li class="nav-item @if(request()->routeIs('admin.messages')) active
@endif">
                <a class="nav-link" href="{{route('admin.messages')}}">Admin
messages</a>
            </li>
            <li class="nav-item ml-lg-auto">
                <span class="text-white">{{session()->get('user')->name}}</span>
                <a class="auth-button" href="{{route('logout')}}">Logout</a>
            </li>
        </ul>
    </div>
</nav>
```

3.7.1.5 scripts.blade.php

```
<script src="https://code.jquery.com/jquery-3.5.1.min.js" integrity="sha256-
9/aliU8dGd2tb6OSsuzixeV4y/faTqgFtohetphbbj0=" crossorigin="anonymous"></script>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-ho+j7jyWK8fNQe+A12Hb8AhRq26LrZ/JpcUGGOn+Y7RsweNrtN/tE3MoK7ZeZDyx"
crossorigin="anonymous"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/toastr.js/latest/toastr.min.js"></script>
<script>
    $.ajaxSetup({
        headers: {
            'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')
        }
    });
    toastr.options = {
        "closeButton": true,
        "positionClass": "toast-bottom-right",
        "timeOut": "5000",
        "hideEasing": "linear",
        "showMethod": "fadeIn",
        "hideMethod": "fadeOut"
```

```

    }
</script>
@stack('scripts')
<script src="{{asset('js/main.js')}}"></script>

```

3.7.2 Layouts

3.7.2.1 admin_layout.blade.php

```

<!DOCTYPE html>
<html>
@include('fixed.head')
@include('fixed.nav_admin')

@yield('content')

@include('fixed.scripts')
</html>

```

3.7.2.2 layout.blade.php

```

<!DOCTYPE html>
<html>
@include('fixed.head')
@include('fixed.nav')

@yield('content')

@include('fixed.footer')
@include('fixed.scripts')
</html>

```

3.7.3 Pages

3.7.3.1 Admin

3.7.3.1.1 index.blade.php

```

@extends('layouts.admin_layout')

@section('keywords')
books, bookstore, admin, novels, home
@endsection

@section('description')
Bookstore - admin home page.
@endsection

@section('title')
Admin dashboard
@endsection

@section('content')
<div class="container p-2 prostor">
    <h1 class="accent mt-2">Admin panel</h1>
    <div class="row w-100">
        <div class="col-12">
            <div class="row w-100">
                <div class="col-12 col-sm-6 mb-2">
                    <h2>Recent user activity</h2>
                </div>
                <div class="col-12 col-sm-6 mb-2">
                    <form action="{{route('admin.home')}}" method="GET">
                        <label for="sortBy">Sort by</label>

```

```

        <select name="sortBy">
            <option {{(isset($qs['sort']) && $qs['sort'] ==
'created_at-asc')} ? 'selected' : ''}} value="created_at-asc">Date ASC</option>
            <option {{(isset($qs['sort']) && $qs['sort'] ==
'created_at-desc')} ? 'selected' : ''}} value="created_at-desc">Date Desc</option>
        </select>
        <input type="submit" value="Sort" class="btn btn-outline-
warning">
    </form>
</div>
</div>
</div>
<div class="col-11 p-0">
    <div class="table-responsive-sm">
        <table class="table">
            <thead>
                <tr>
                    <th>Activity</th>
                    <th>User name</th>
                    <th>Date</th>
                </tr>
            </thead>
            <tbody>
                @forelse($activities as $a)
                    <tr>
                        <td>{{ $a->type }}</td>
                        <td>{{ $a->user_name }}</td>
                        <td>{{ $a->created_at }}</td>
                    </tr>
                @empty
                    <tr>
                        <td>No recent activities to display</td>
                    </tr>
                @endforelse
            </tbody>
        </table>
    </div>
</div>
    {{ $activities->links("pagination::bootstrap-4") }}
</div>
</div>
@endsection

```

3.7.3.1.2 messages.blade.php

```

@extends('layouts.admin_layout')

@section('keywords')
    books, bookstore, admin, novels, home
@endsection

@section('description')
    Bookstore - admin home page.
@endsection

@section('title')
    Admin dashboard
@endsection

@section('content')
    <div class="container prostor">
        <div class="row my-3 justify-content-center">
            @forelse($messages as $m)
                <div class="col-8 border m-2 mx-auto">

```

```

                <h2>User message</h2>
                <p>Sent by: {{$m->name}}</p>
                <p>Email: {{$m->email}}</p>
                <p>Message: </p>
                <p>{{$m->message}}</p>
                <p>Date received: {{$m->created_at}} </p>
            </div>
        @empty
            <div class="mt-5 col-10 text-center alert alert-danger"><p>There are no
messages to display.</p></div>
        @endforelse

    </div>
</div>
@endsection

```

3.7.3.1.3 Books

3.7.3.1.3.1 create.blade.php

```

@extends('layouts.admin_layout')

@section('keywords')
    books, bookstore, admin, novels, add
@endsection

@section('description')
    Bookstore - admin add new book
@endsection

@section('title')
    Add a new book
@endsection

@section('content')
    <div class="container p-3 prostor">
        <div class="mx-auto">
            @if(session()->has('success'))
                <div class="alert alert-success rounded text-center">
                    <p>Added new book!</p>
                </div>
            @endif
            @if(session()->has('error'))
                <div class="alert alert-danger rounded text-center">
                    <p>{{session()->get('error')}}</p>
                </div>
            @endif
            <h1 class="accent">Add a new book</h1>
        </div>
        <div class="row p-2">
            <div class="col-12">
                <form action="{{route('books.store')}}" method="POST"
enctype="multipart/form-data">
                    <div class="form-group">
                        <label for="title">Title</label>
                        <input type="text" class="form-control" name="title"
placeholder="Book title" value="{{old('title')}}">
                        <span class="text-
danger">@error('title') {{$message}}@enderror</span>
                    </div>
                    <div class="form-group">
                        <label for="genreId">Select genre</label>
                        <select class="form-control" name="genreId">

```

```

        <option value="0">Select</option>
        @foreach($genres as $g)
        @if(old('genreId') == $g->id)
            <option selected value="{{ $g->id }}">{{ $g-
>genre_name}}</option>
        @else
            <option value="{{ $g->id }}">{{ $g->genre_name}}</option>
        @endif
        @endforeach
    </select>
    <span class="text-
danger">@error('genreId'){{ $message}}@enderror</span>
</div>
<div class="form-group container1">
    <label>Author</label>
    <input type="text" class="form-control mb-2" name="author1"
value="{{ old('author1') }}" placeholder="Author name">
    <input type="text" class="form-control mb-2" name="author2"
value="{{ old('author2') }}" placeholder="Optional">
    <span class="text-
danger">@error('author1'){{ $message}}@enderror</span>
    <span class="text-
danger">@error('author2'){{ $message}}@enderror</span>

</div>
<div class="form-group">
    <label for="price">Price</label>
    <input type="text" class="form-control"
value="{{ old('price') }}" name="price" placeholder="Price"/>
    <span class="text-
danger">@error('price'){{ $message}}@enderror</span>
</div>
<div class="form-group">
    <label for="publishYear">Publication year</label>
    <input type="text" class="form-control"
value="{{ old('publishYear') }}" name="publishYear" placeholder="Publication year"/>
    <span class="text-
danger">@error('publishYear'){{ $message}}@enderror</span>
</div>
<div class="form-group">
    <label for="description">Description</label>
    <textarea class="form-control" name="description"
rows="3">{{ old('description') }}</textarea>
    <span class="text-
danger">@error('description'){{ $message}}@enderror</span>
</div>
<div class="form-group">
    <label for="image">Image</label>
    <input type="file" name="image" class="form-control-file"/>
    <span class="text-
danger">@error('image'){{ $message}}@enderror</span>
</div>
<div class="form-group d-flex justify-content-center">
    <div><input type="submit" value="Add" class="btn btn-outline-
warning"/></div>

</div>
@csrf
</form>
</div>
</div>
</div>
@endsection

```

3.7.3.1.3.2 edit.blade.php

```
@extends('layouts.admin_layout')

@section('keywords')
    books, bookstore, admin, novels, edit
@endsection

@section('description')
    Bookstore - admin edit book.
@endsection

@section('title')
    Edit book
@endsection

@section('content')
    <div class="container p-3 prostor">
        <div class="mx-auto">
            @if(session()->has('error'))
                <div class="alert alert-danger rounded text-center">
                    <p>{{session()->get('error')}}</p>
                </div>
            @endif
            <h1 class="accent">Edit book</h1>
        </div>
        <div class="row p-2">
            <div class="col-12">
                <form action="{{route('books.update', ['book'=>$book->id])}}"
method="POST" enctype="multipart/form-data">
                    <div class="form-group">
                        <label for="title">Title</label>
                        <input type="text" class="form-control" name="title"
placeholder="Book title" value="{{ $book->book_name }}">
                        <span class="text-
danger">@error('title') {{ $message }} @enderror</span>
                    </div>
                    <div class="form-group">
                        <label for="genreId">Select genre</label>
                        <select class="form-control" name="genreId">
                            <option value="0">Select</option>
                            @foreach($genres as $g)
                                @if($book->genre->id == $g->id)
                                    <option selected value="{{ $g->id }}">{{ $g-
>genre_name }}</option>
                                @else
                                    <option value="{{ $g->id }}">{{ $g-
>genre_name }}</option>
                                @endif
                            @endforeach
                        </select>
                        <span class="text-
danger">@error('genreId') {{ $message }} @enderror</span>
                    </div>
                    <div class="form-group">
                        <label>Author</label>
                        @if(count($book->authors) == 1)
                            <input type="text" class="form-control mb-2" name="author1"
value="{{ $book->authors[0]->name }}" placeholder="Author name">
                            <span class="text-
danger">@error('author1') {{ $message }} @enderror</span>
                        @else
                            <input type="text" class="form-control mb-2" name="author1"
value="{{ $book->authors[0]->name }}" placeholder="Author name">
                            <input type="text" class="form-control mb-2" name="author2"
value="{{ $book->authors[1]->name }}" placeholder="Author name">
                        @endif
                    </div>
                </form>
            </div>
        </div>
    </div>
```



```

        <span class="text-
danger">@error('author1'){{ $message}}@enderror</span>
        <span class="text-
danger">@error('author2'){{ $message}}@enderror</span>
        @endif
    </div>
    <div class="form-group">
        <label for="price">Price</label>
        <input type="text" class="form-control" value="{{ $book-
>price}}" name="price" placeholder="Price"/>
        <span class="text-
danger">@error('price'){{ $message}}@enderror</span>
    </div>
    <div class="form-group">
        <label for="publishYear">Publication year</label>
        <input type="text" class="form-control" value="{{ $book-
>publish_year}}" name="publishYear" placeholder="Publication year"/>
        <span class="text-
danger">@error('publishYear'){{ $message}}@enderror</span>
    </div>
    <div class="form-group">
        <label for="description">Description</label>
        <textarea class="form-control" name="description"
rows="3">{{ $book->description}}</textarea>
        <span class="text-
danger">@error('description'){{ $message}}@enderror</span>
    </div>
    <div class="form-group">
        <label for="image">Image</label>
        <input type="file" name="image" class="form-control-file"/>
        <span class="text-
danger">@error('image'){{ $message}}@enderror</span>
    </div>
    <div class="form-group d-flex justify-content-center">
        <div><input type="submit" value="Edit" class="btn btn-outline-
warning"/></div>
    </div>
    @csrf
    @method('put')
</form>
</div>
</div>
</div>
@endsection

```

3.7.3.1.3.3 products.blade.php

```

@extends('layouts.admin_layout')

@section('keywords')
    books,bookstore,admin,novels,products
@endsection

@section('description')
    Bookstore - admin view all products.
@endsection

@section('title')
    Admin products
@endsection

@php

$sortBy=[['key'=>'Price','value'=>'ASC'],['key'=>'Price','value'=>'DESC'],['key'=>'book
_name','value'=>'ASC'],['key'=>'book_name','value'=>'DESC']];

```

```

@endphp
@section('content')
    <div class="container mt-5 prostor">
        @if(session()->has('success'))
            <div class="alert alert-success rounded text-center">
                <p>{{session()->get('success')}}</p>
            </div>
        @endif
        @if(session()->has('error'))
            <div class="alert alert-danger rounded text-center">
                <p>{{session()->get('error')}}</p>
            </div>
        @endif
        <div class="row w-100 m-0">
            <div class="col-lg-3 p-2">
                <div class="row">
                    <div class="col-12">
                        <form action="{{route('admin.products')}}" method="GET">
                            <p class="font-weight-bold accent">Sort by:</p>
                            <select class="custom-select" name="sortBy">
                                @foreach($sortBy as $s)
                                    @if(isset($qs['sortBy']) && explode('-', $qs['sortBy'])[0] == $s['key'] && explode('-', $qs['sortBy'])[1] == $s['value'])
                                        <option selected value="{{ $s['key'] }}"> {{ucfirst(implode(' ', explode('_', $s['key'])))}}
                                        {{ $s['value'] }}</option>
                                    @else
                                        <option value="{{ $s['key'] }}-{{ $s['value'] }}">
                                        {{ucfirst(implode(' ', explode('_', $s['key'])))}} {{ $s['value'] }}</option>
                                    @endif
                                @endforeach
                            </select>
                            <br/><br/>
                            <ul class="list-group">
                                <li class="list-group-item">
                                    <label for="pretraga" class="accent">Search</label>
                                    <input type="text" class="form-control"
name="search" value="{{ isset($qs["search"]) ? $qs["search"] : "" }}" />
                                </li>
                            </ul>
                            <br/>
                            <div class="row">
                                <div class="col-12">
                                    <p class="font-weight-bold accent">Genres:</p>
                                    <ul class="list-group" id="genres">
                                        @foreach($genres as $g)
                                            <li class="list-group-item">
                                                <input type="checkbox" {{
(isset($qs["chbGenres"]) && in_array($g->id, $qs["chbGenres"])) ? "checked" : "" }}
value="{{ $g->id }}" name="chbGenres[]" /> {{ $g->genre_name }} ({{ count($g->books) }})
                                            </li>
                                        @endforeach
                                    </ul>
                                    <br/>
                                    <input type="submit" value="Search" class="btn btn-
outline-warning">
                                </div>
                            </div>
                        </form>
                    </div>
                </div>
            </div>
            <div class="col-lg-9 py-2">
                <h1 class="accent">Books ({{ $books->total() }})</h1>
            </div>
        </div>
    </div>

```

```

        <a href="{{route('books.create')}}" class="btn btn-primary">Add a new
book</a>

        <div class="row m-0 w-100">
            @forelse($books as $b)
                @include('pages.partials.product')
            @empty
                <div class="container not-found m-5 alert alert-
danger"><p>There are no products that match the criteria.</p></div>
            @endforelse
        </div>
    </div>
    <div class="col-9 offset-3" id="pagination">
        {{ $books->links("pagination::bootstrap-4") }}
    </div>
</div>
</div>
@endsection

```

3.7.3.1.4 Genres

3.7.3.1.4.1 create.blade.php

```

@extends('layouts.admin_layout')

@section('keywords')
    books, bookstore, admin, genre, new
@endsection

@section('description')
    Bookstore - admin add a new genre.
@endsection

@section('title')
    Admin add genre
@endsection

@section('content')
    <div class="container p-3 prostor">
        <div class="mx-auto">
            @if(session()->has('success'))
                <div class="alert alert-success rounded text-center">
                    <p>{{session()->get('success')}}</p>
                </div>
            @endif
            @if(session()->has('error'))
                <div class="alert alert-danger rounded text-center">
                    <p>{{session()->get('error')}}</p>
                </div>
            @endif
            <h1 class="accent">Add a new genre</h1>
        </div>
        <div class="row p-2">
            <div class="col-12">
                <form action="{{route('admin.genres.store')}}" method="POST">
                    <div class="form-group">
                        <label for="genreName">Genre name</label>
                        <input type="text" class="form-control" name="genreName"
placeholder="Genre name" value="{{old('genreName')}}">
                        <span class="text-
danger">@error('genreName') {{ $message }} @enderror</span>
                    </div>
                    <div class="form-group d-flex justify-content-center">
                        <div><input type="submit" value="Add" class="btn btn-outline-
warning"/></div>
                </form>
            </div>
        </div>
    </div>

```

```

        </div>
        @csrf
    </form>
</div>
</div>
</div>
@endsection

```

3.7.3.1.4.2 edit.blade.php

```

@extends('layouts.admin layout')

@section('keywords')
    books, bookstore, admin, genre, new
@endsection

@section('description')
    Bookstore - admin add a new genre.
@endsection

@section('title')
    Admin add genre
@endsection

@section('content')
    <div class="container p-3 prostor">
        <div class="mx-auto">
            @if(session()->has('success'))
                <div class="alert alert-success rounded text-center">
                    <p>{{session()->get('success')}}</p>
                </div>
            @endif
            @if(session()->has('error'))
                <div class="alert alert-danger rounded text-center">
                    <p>{{session()->get('error')}}</p>
                </div>
            @endif
            <h1 class="accent">Edit genre</h1>
        </div>
        <div class="row p-2">
            <div class="col-12">
                <form action="{{route('admin.genres.update', ['genre'=>$genre->id])}}"
method="POST">
                    <div class="form-group">
                        <label for="genreName">Genre name</label>
                        <input type="text" class="form-control" name="genreName"
placeholder="Genre name" value="{{ $genre->genre_name }}">
                        <span class="text-
danger">@error('genreName') {{ $message }} @enderror</span>
                    </div>
                    <div class="form-group d-flex justify-content-center">
                        <div><input type="submit" value="Edit" name="btnAddGenre"
class="btn btn-outline-warning"/></div>
                    </div>
                    @csrf
                    @method('put')
                </form>
            </div>
        </div>
    </div>
@endsection

```

3.7.3.1.4.3 index.blade.php

```
@extends('layouts.admin_layout')

@section('keywords')
    books,bookstore,admin,novels,genres
@endsection

@section('description')
    Bookstore - admin genre page.
@endsection

@section('title')
    Admin genres
@endsection

@section('content')
    <div class="container prostor pt-3">
        <div class="mx-auto my-3">
            @if(session()->has('success'))
                <div class="alert alert-success rounded text-center">
                    <p>{{session()->get('success')}}</p>
                </div>
            @endif
            @if(session()->has('error'))
                <div class="alert alert-danger rounded text-center">
                    <p>{{session()->get('error')}}</p>
                </div>
            @endif
            <h1 class="accent">Genres</h1>
            <a href="{{route('admin.genres.create')}}" class="btn btn-primary">Add new
genre</a>
        </div>
        <div>
            <table class="table">
                <thead>
                    <tr>
                        <th>Name</th>
                        <th>Edit</th>
                        <th>Delete</th>
                    </tr>
                </thead>
                <tbody>
                    @foreach($genres as $g)
                        <tr>
                            <td>{{ $g->genre_name }}</td>
                            <td><a href="{{route('admin.genres.edit',['genre'=>$g->id])}}"
class="btn btn-outline-warning">Edit</a></td>
                            <td>
                                <form action="{{route('admin.genres.delete',['genre'=>$g-
>id])}}" method="POST">
                                    <input type="submit" class="btn btn-outline-danger"
value="Delete"/>
                                    @csrf
                                    @method('delete')
                                </form>
                            </td>
                        </tr>
                    @endforeach
                </tbody>
            </table>
        </div>
    </div>
@endsection
```

3.7.3.1.5 Users

3.7.3.1.5.1 index.blade.php

```
@extends('layouts.admin_layout')

@section('keywords')
    books,bookstore,admin,novels,users
@endsection

@section('description')
    Bookstore - admin users page.
@endsection

@section('title')
    Admin users
@endsection

@section('content')
    <div class="container prostor pt-3">
        <div class="mx-auto my-3">
            @if(session()->has('success'))
                <div class="alert alert-success rounded text-center">
                    <p>{{session()->get('success')}}</p>
                </div>
            @endif
            @if(session()->has('error'))
                <div class="alert alert-danger rounded text-center">
                    <p>{{session()->get('error')}}</p>
                </div>
            @endif
            <h1 class="accent">Users</h1>
        </div>
        <div>
            <table class="table">
                <thead>
                    <tr>
                        <th>User ID</th>
                        <th>Name</th>
                        <th>Email</th>
                        <th>Role</th>
                        <th>Delete</th>
                    </tr>
                </thead>
                <tbody>
                    @foreach($users as $u)
                        @if($u->role->role_name != 'admin')
                            <tr>
                                <td>{{ $u->id }}</td>
                                <td>{{ $u->name }}</td>
                                <td>{{ $u->email }}</td>
                                <td>{{ $u->role->role_name }}</td>
                                <td>
                                    <form action="{{ route('admin.users.delete', ['user'=>$u->id]) }}" method="POST">
                                        <input type="submit" class="btn btn-outline-danger"
                                        value="Delete"/>
                                        @csrf
                                        @method('delete')
                                    </form>
                                </td>
                            </tr>
                        @endif
                    @endforeach
                </tbody>
            </table>
        </div>
    </div>
endsection
```

```

        </div>
    </div>
@endsection

```

3.7.3.2 Auth

3.7.3.2.1 login.blade.php

```

@extends('layouts.layout')

@section('keywords')
    books, bookstore, account, novels, login
@endsection

@section('description')
    Bookstore - login to your account.
@endsection

@section('title')
    Login
@endsection

@section('content')
    <div class="prostor">
        <div class="container py-5">
            <h1 class="text-center accent">Login</h1>
            <div class="row py-5">
                <div class="col-12 col-lg-5 mx-auto py-5">
                    @if(session()->has('error'))
                        <div class="alert alert-danger rounded text-center">
                            <p>{{session()->get('error')}}</p>
                        </div>
                    @endif
                    <form action="{{route('doLogin')}}" name="logForm" id="logForm"
method="post">
                        <div class="form-group">
                            <label for="tbEmail">Email</label>
                            <input type="text" name="tbEmail" id="tbEmail" class="form-
control" value="{{old('tbEmail')}}"/>
                            <p class="greska text-danger">@error('tbEmail')
{{ $message }} @enderror</p>
                        </div>
                        <div class="form-group">
                            <label for="tbPass">Password</label>
                            <input type="password" name="tbPass" id="tbPass"
class="form-control" />
                            <p class="greska text-danger">@error('tbPass') {{ $message }}
@enderror</p>
                        </div>
                        <div class="form-group text-center">
                            <input type="submit" name="btnLogin" id="btnLogin"
class="auth-button" value="Login" />
                            <p class="uspeh text-primary"></p>
                        </div>
                        @csrf
                    </form>
                </div>
            </div>
        </div>
    </div>
@endsection

```

3.7.3.2.2 register.blade.php

```
@extends('layouts.layout')

@section('keywords')
    books, bookstore, account, novels, register
@endsection

@section('description')
    Bookstore - register and create a new account.
@endsection

@section('title')
    Register
@endsection

@section('content')
    <div class="prostor">
        <div class="container py-5">
            <h1 class="text-center accent">Register</h1>
            <div class="row py-5">
                <div class="col-12 col-lg-5 mx-auto py-5">
                    @if(session()->has('error'))
                        <div class="alert alert-danger rounded text-center">
                            <p>{{session()->get('error')}}</p>
                        </div>
                    @endif
                    @if(session()->has('success'))
                        <div class="alert alert-success rounded text-center">
                            <p>{{session()->get('success')}}</p><span>Click <a
class='accent' href="{{route('login')}}">here</a> to login now</span>
                        </div>
                    @endif
                    <form action="{{route('doRegister')}}" name="regForm" id="regForm"
method="post">
                        <div class="form-group">
                            <label for="tbName">Full name<mark>*</mark></label>
                            <input type="text" name="tbName" id="tbName" class="form-
control" value="{{old('tbName')}}"/>
                            <p class="greska text-danger">@error('tbName') {{$message}}
@enderror</p>
                        </div>
                        <div class="form-group">
                            <label for="tbEmail">Email<mark>*</mark></label>
                            <input type="text" name="tbEmail" id="tbEmail" class="form-
control" value="{{old('tbEmail')}}"/>
                            <p class="greska text-danger">@error('tbEmail')
{{$message}} @enderror</p>
                        </div>
                        <div class="form-group">
                            <label for="tbPass">Password<mark>*</mark></label>
                            <input type="password" name="tbPass" id="tbPass"
class="form-control" />
                            <p class="greska text-danger">@error('tbPass') {{$message}}
@enderror</p>
                        </div>
                        <div class="col-12"><p><mark>*</mark> Required fields</p></div>
                        <div class="form-group text-center">
                            <input type="submit" name="btnReg" id="btnReg" class="auth-
button" value="Register" />
                            <p class="uspeh text-primary"></p>
                        </div>
                        @csrf
                    </form>
                </div>
            </div>
        </div>
    </div>
```



```

        </div>
    </div>
@endsection

```

3.7.3.3 Books

3.7.3.3.1 cart.blade.php

```

@extends('layouts.layout')

@section('keywords')
    books, bookstore, cart, novels, order
@endsection

@section('description')
    Bookstore - review the items in your cart.
@endsection

@section('title')
    Cart
@endsection

@section('content')
    <div class="prostor">
        <div class="container mx-auto mt-5">
            @if(count($cartItems) == 0)
                @if(session()->has('msg'))
                    <div class=" mt-5 mx-auto alert alert-success"><p>Your order has
been placed. Go back to the <a href="{{route('books.index')}}">store</a></p></div>
                @else
                    <div class=" mt-5 mx-auto alert alert-danger"><p>You have nothing in
your cart. Visit the <a href="{{route('books.index')}}">store</a></p></div>
                @endif
            @else
                <div class="row">
                    <div class="container">
                        <h1 class="accent text-center"><b>Shopping Cart</b></h1>
                    </div>

                    <div class="col-12 cart">
                        <div class="title">
                            <div class="row">
                                <div class="col align-self-center text-right accent"
id="count">{{count($cartItems)}} items</div>
                                </div>
                            </div>

                            @foreach($cartItems as $c)
                                <div class="border-top border-bottom p-2 mb-3 cart-item"
id="cart_item_{{ $c->bookId }}">
                                    <div class="row main align-items-center justify-
content-between">
                                        <div class="col-3 col-md-2">image) }}"></div>
                                        <div class="col-3 col-md-2">
                                            <div class="row accent">Book</div>
                                            <div class="row"><a id="name"
href="{{route('books.show', ['book'=>$c->bookId]) }}">{{ $c->name }}</a></div>
                                            </div>
                                            <div class="col-2">
                                                <div class="row accent">Quantity</div>
                                                <div class="row"><input class="form-control
quantity" min="1" data-price="{{ $c->price }}" data-id="{{ $c->bookId }}" type="number"
value="{{ $c->quantity }}"></div>
                                            </div>
                                        </div>
                                    </div>
                                </div>
                            @endforeach
                        </div>
                    </div>
                </div>
            @endif
        </div>
    </div>

```

```

                <div class="col-2">
                    <div class="row accent">Price</div>
                    <div class="row">&euro;<span
class="product_price" id="span_price_{{ $c->bookId }}">{{ $c->price * $c->quantity
}}</span></div>
                </div>
                <div class="col-1">
                    <a href="#"><span class="remove" data-
id="{{ $c->bookId }}" class="close">&#10005;</span></a>
                </div>
            </div>
        </div>
        @endforeach

        <div class="back-to-shop"><a href="/books">&leftarrow;<span
class="text-muted">Back to shop</span></a></div>
        <div class="col text-right"><span class="accent font-weight-
bold">Total price: </span>&euro; <span id="totalPrice"> </span></div>
        <div class="col text-right"><a href="{{ route('checkout') }}"
class="btn btn-outline-warning" id="checkout">Checkout</a></div>
    </div>
</div>
@endif
<br>
<br>
<br>
</div>
</div>
@endsection

```

3.7.3.3.2 index.blade.php

```

@extends('layouts.layout')

@section('keywords')
    books, bookstore, all, novels, search
@endsection

@section('description')
    Bookstore - see our offer of books.
@endsection

@section('title')
    Books
@endsection

@section('content')
    <div class="container mt-5 prostor">
        <div class="row w-100 m-0">
            <div class="col-lg-3 p-2">
                <div class="row">
                    <div class="col-12">
                        <p class="font-weight-bold accent">Sort by:</p>
                        <select class="custom-select" name="sortBy" id="sort">
                            <option value="price-asc">Price asc</option>
                            <option value="price-desc">Price desc</option>
                            <option value="book_name-asc">Book name asc</option>
                            <option value="book_name-desc">Book name desc</option>
                        </select>
                        <br/><br/>
                        <ul class="list-group">
                            <li class="list-group-item">
                                <label for="pretraga" class="accent">Search</label>
                                <input type="text" class="form-control" id="search"

```

```

name="search"/>
        </li>
    </ul>
    <br/>

    <div class="row">
        <div class="col-12">
            <p class="font-weight-bold accent">Genres:</p>
            <ul class="list-group" id="genres">
                @foreach($genres as $g)
                    <li class="list-group-item">
                        <input type="checkbox" value="{{ $g->id }}"
class="genre" name="genres"/> {{ $g->genre_name}} ({{count($g->books)}})
                    </li>
                @endforeach
            </ul>
            <br/>
        </div>
    </div>
</div>
</div>
</div>
<div class="col-lg-9 py-2">
    <h1 class="accent" id="count">Books</h1>
    <div class="row m-0 w-100" id="products">

    </div>
    <div id="loader" class="justify-content-center">
        <div class="loadingio-spinner-rolling-5dgev8yz7mg"><div
class="ldio-zftvpl388ch">
            <div></div>
        </div></div>
    </div>
</div>
<div class="col-9 offset-3" id="pagination">
    <ul class="pagination">

    </ul>
</div>
</div>
</div>
@endsection
@push('scripts')
<script>
    const imgPath = '{{asset('img/books/')}}';
</script>
@endpush

```

3.7.3.3 show.blade.php

```

@extends('layouts.layout')

@section('keywords')
    books,bookstore,book,novels,information
@endsection

@section('description')
    Bookstore - see book details.
@endsection

@section('title')
    {{ $book->book_name}}
@endsection

```

```
@section('content')
    @include('pages.partials.singleProduct')
@endsection
```

3.7.3.4 Partial

3.7.3.4.1 product.blade.php

```
@php
    $authorArr = [];
    foreach($b->authors as $a){
        array_push($authorArr,$a->name);
    }
    $author = implode(' ', $authorArr);
@endphp


#### 3.7.3.4.2 singleProduct.blade.php



```
@php
 $authorArr = [];
 foreach($book->authors as $a){
 array_push($authorArr,$a->name);
 }
 $author = implode(' ', $authorArr);
@endphp

84


```


```

```

        <p>Genre: {{ $book->genre->genre_name }}</p>
        <p>Publication year: {{ $book->publish_year }}</p>
        <p>Description: </p>
        <p>{{ $book->description }}</p>
        <p>Price: ${{ $book->price }}</p>
        @if(session()->has('user') && session()->get('user')->role->role_name ==
'admin')
            <a href="{{ route('books.edit', ['book'=>$book->id]) }}" class="btn btn-
outline-warning mt-2">Edit</a>
            <form action="{{ route('books.destroy', ['book'=>$book->id]) }}"
method="POST">
                <input type="submit" class="btn btn-outline-danger mt-2"
value="Delete"/>
                @csrf
                @method('delete')
            </form>
        @else
            <a href="#" class="btn btn-outline-warning mt-2 addToCart" data-
id="{{ $book->id }}">Add to cart</a>
        @endif
    </div>
</div>
</div>

```

3.7.3.5 *author.blade.php*

```
@extends('layouts.layout')

@section('keywords')
    books,bookstore,author,novels,information
@endsection

@section('description')
    Bookstore - information about the author of the site.
@endsection

@section('title')
    Author
@endsection

@section('content')
    <div class="prostor">
        <h1 class="text-center mt-3 accent">Information about the site author</h1>
        <div class="container mx-auto mt-5 row">
            <div class="col-11 col-md-5 p-4">
                
            </div>
            <div class="col-12 col-md-7 p-5">
                <h4>Djordje Solomun 16/19</h4>
                <p>I am a third year student at the Department School of Information
and Communication Technologies, Academy of Technical and Art Applied Studies Belgrade.
This website is a project for the Web programming PHP 2 course.</p>
                <p>If you have any questions use my email to reach me:
djordje&#46;solomun&#46;16&#46;19&#64;ict&#46;edu&#46;rs</p>
                <p>Take a look at my <a href="https://djotaa.github.io/WebPortfolio/"
target="blank">portfolio</a></p>
            </div>
        </div>
    </div>
@endsection
```

3.7.3.6 contact.blade.php

```
@extends('layouts.layout')

@section('keywords')
    books,bookstore,contact,novels,information
@endsection

@section('description')
    Bookstore - if you have a question, please contact us.
@endsection

@section('title')
    Contact us
@endsection

@section('content')
    <div class="prostor">
        <h1 class="text-center my-5 accent">Contact us</h1>
        <div class="container mx-auto row">
            <div class="col-12 col-md-6 mb-2 p-3">
                <h3 class="text-center accent">Send us a message</h3>
                <form method="post" action='{{route('contact.send')}}' id="con-form"
name="conForm">
                    <div class="row">
                        <div class="form-group col-12">
                            <label for="tbName">Full name<mark>*</mark></label>
                            <input type="text" class="form-control" name="tbName"
id="tbName"placeholder="ex. John Doe" value="{{old('tbName')}}"/>
                            <p class="greska text-danger">@error('tbName') {{$message}}
@enderror</p>
                        </div>
                        <div class="form-group col-12">
                            <label for="tbEmail">Email<mark>*</mark></label>
                            <input class="form-control" type="text" name="tbEmail"
id="tbEmail" placeholder="ex. johndoe@gmail.com" value="{{old('tbEmail')}}"/>
                            <p class="greska text-danger">@error('tbEmail')
{{$message}} @enderror</p>
                        </div>
                        <div class="form-group col-12">
                            <label for="tbMessage">Your message<mark>*</mark></label>
                            <textarea name="userMessage" id="userMessage" class="form-
control" placeholder="Your message.">{{old('tbMessage')}}</textarea>
                            <p class="greska text-danger">@error('userMessage')
{{$message}} @enderror</p>
                        </div>
                        <div class="col-12"><p><mark>*</mark> Required fields</p></div>
                        <div class="form-group col">
                            <input type="submit" value="Send" name="submitContact"
id="btnSubmitContact" class="btn btn-outline-warning d-block mx-auto"/>
                            @if(session()->has('success'))
                                <p class="text-primary"> {{ session()->get('success')}
}}</p>
                            @elseif(session()->has('error'))
                                <p class="text-danger"> {{ session()->get('error')}
}}</p>
                            @endif
                        </div>
                    </div>
                </div>
            </div>
            @csrf
        </form>
    </div>
    <div class="col-12 col-md-6 p-3">
        <h3 class="text-center accent">Our contact information</h3>
        <ul id="kontakt">
            <li class="my-4 accent"><i class="fas fa-map-marker-alt mr-
```

```

3"></i>Knez Mihailova 10, Belgrade</li>
        <li class="my-4 accent"><i class="fas fa-phone mr-3"></i>+381 64
987654</li>
        <li class="my-4 accent"><i class="fas fa-phone mr-3"></i>+381 11
234567</li>
        <li class="my-4 accent"><i class="far fa-envelope mr-
3"></i>bookstore@gmail.com</li>
    </ul>
</div>
</div>
</div>
@endsection

```

3.7.3.7 [home.blade.php](#)

```

@extends('layouts.layout')

@section('keywords')
    books,bookstore,bookshop,novels,home
@endsection

@section('description')
    Bookstore - buy books of all genres and authors.
@endsection

@section('title')
    Home
@endsection

@section('content')
    <div class="container mt-3">
        <div id="carouselExampleFade" class="carousel slide carousel-fade" data-
ride="carousel">
            <div class="carousel-inner rounded">
                <div class="carousel-item active" data-interval="3000">
                    
                    <div class="carousel-caption rounded d-none d-md-block text-body">
                        <h1 class="accent">Bookstore</h1>
                        <p>Online book shop</p>
                    </div>
                </div>
                <div class="carousel-item" data-interval="3000">
                    
                    <div class="carousel-caption rounded d-none d-md-block text-body">
                        <h2 class="accent">Everything you need</h2>
                        <p>Find books from your favorite authors and genres. Check out
the <a href="{{route('books.index')}}" class="btn btn-outline-warning">shop</a></p>
                    </div>
                </div>
            </div>
        </div>
    </div>
    <div class="container mt-5">
        <h2 class="text-center accent">PC Center</h2>
        <div class="row mt-3 d-flex align-items-center">
            <div class="col-9 col-md-6 text-center mx-auto">
                <p>If you like to read this is the place for you. We have something for
everyone, and we are always expanding our offer.</p>
                <p>Take a look at our book collection: <a
href="{{route('books.index')}}" class="btn btn-outline-warning">shop</a></p>
            </div>
            <div class="col-9 col-md-6 mx-auto">

```

```

        
    </div>
</div>
</div>
<div class="container mt-5">
    <h2 class="text-center accent">Newest releases</h2>
    <div class="row w-100 mt-3 pb-3 d-flex justify-content-center align-items-
center">
        @foreach($books as $b)
            @php
                $authoriN = [];
                foreach($b->authors as $a){
                    array_push($authoriN,$a->name);
                }
                $authori = implode(', ', $authoriN);
            @endphp

            <div class="col-10 col-sm-6 col-md-3 mt-2 mx-auto">
                <div class="product card">
                    
                    <div class="card-body">
                        <a href="{{route('books.show', ['book'=>$b->id])}}"
id="name">
                            <h5 class="card-title">{{ $b->book_name }}</h5><hr>
                        </a>
                        <p class="card-text m-0">Genre: {{ $b->genre-
>genre_name }}</p><hr>
                        <p class="card-text m-0">Author: {{ $authori }}</p><hr>
                        <p class="card-text font-weight-bold m-0" id="price">Price:
${{ $b->price }}</p>
                        <a href="#" class="btn btn-outline-warning mt-2 addToCart"
id="1">Add to cart</a>
                    <p></p>
                    </div>
                </div>
            </div>
        @endforeach
    </div>
</div>
@endsection

```

3.8 Routes

3.8.1 web.php

```

<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\AuthController;
use App\Http\Controllers\HomeController;
use App\Http\Controllers>ContactController;
use App\Http\Controllers\BookController;
use App\Http\Controllers\AdminController;
use App\Http\Controllers\CartController;
use App\Http\Controllers\GenreController;

Route::redirect('/', '/home');
Route::get('/home', [HomeController::class, 'index'])->name('home');
Route::get('/about', [HomeController::class, 'about'])->name('about');
Route::get('/contact', [ContactController::class, 'index'])->name('contact.index');
Route::post('/contact', [ContactController::class, 'send'])->name('contact.send');
Route::get('/author', [HomeController::class, 'author'])->name('author');

```



```

Route::middleware('isGuest')->group(function() {
    Route::get('/login', [AuthController::class, 'loginForm'])->name('login');
    Route::post('/login', [AuthController::class, 'login'])->name('doLogin');
    Route::get('/register', [AuthController::class, 'registerForm'])->name('register');
    Route::post('/register', [AuthController::class, 'register'])->name('doRegister');
});

Route::middleware('isLoggedIn')->group(function() {
    Route::get('/logout', [AuthController::class, 'logout'])->name('logout');

    Route::get('/cart', [CartController::class, 'get'])->name('cart');
    Route::post("/cart/adjustQuantity", [CartController::class, "adjustQuantity"]);
    Route::delete("/cart/{bookId}", [CartController::class, "remove"]);
    Route::get('/cart/checkout', [CartController::class, 'checkout'])->name('checkout');

    Route::middleware('isAdmin')->group(function() {
        Route::get('/admin', [AdminController::class, 'index'])->name('admin.home');

        Route::get('/admin/products', [AdminController::class, 'products'])->name('admin.products');

        Route::get('/admin/users', [AdminController::class, 'users'])->name('admin.users');
        Route::delete('admin/users/{user}', [AdminController::class, 'destroy_user'])->name('admin.users.delete');

        Route::get('/admin/genres', [GenreController::class, 'index'])->name('admin.genres');
        Route::get('admin/genres/{genre}/edit', [GenreController::class, 'edit'])->name('admin.genres.edit');
        Route::get('admin/genres/create', [GenreController::class, 'create'])->name('admin.genres.create');
        Route::post('admin/genres', [GenreController::class, 'store'])->name('admin.genres.store');
        Route::put('admin/genres/{genre}', [GenreController::class, 'update'])->name('admin.genres.update');
        Route::delete('admin/genres/{genre}', [GenreController::class, 'destroy'])->name('admin.genres.delete');

        Route::get('/admin/messages', [AdminController::class, 'messages'])->name('admin.messages');
    });
});

Route::post('/cart', [CartController::class, 'index']);
Route::get('/books/json', [BookController::class, 'get_json']);
Route::resource('books', BookController::class);

```

3.9 CSS

3.9.1 style.css

```

* {
    padding: 0;
    margin: 0;
}

.carousel-caption {
    background-color: rgba(255, 255, 255, 0.9);
}

.card {
    border: 0px solid black !important;
}

```

```

background-color: transparent;
width: inherit !important;
}

.card:hover {
  box-shadow: 0 0 0 0.2rem rgb(206, 120, 82, 25%);
  transition: 0.5s;
}

ul {
  list-style-type: none;
}

footer p {
  margin-bottom: 0rem !important;
}

footer ul li a {
  color: #fff;
}

footer ul li a:hover, footer h4 {
  color: #ce7852;
}

.prostor {
  min-height: 80vh;
}

#kontakt {
  text-align: center;
}

#kontakt i {
  font-size: 1.5em;
}

form mark{
  color: red;
  font-weight: bold;
  background-color: transparent;
}

#price{
  color: #ce7852;
}

.btn-outline-warning{
  color: #ce7852!important;
  border-color: #ce7852!important;
}

.btn-outline-warning:hover{
  background-color: #ce7852!important;
  color:#fff!important;
}

#name{
  color: #212529;
}

#name:hover{
  text-decoration-color: #ce7852;
}

#loader{
  display:flex;
}

.page-link{
  color: #ce7852;
  border-color: #ce7852;
}

```

```

}
.page-item.active .page-link{
  color: #fff;
  background-color: #ce7852;
  border-color: #ce7852;
}
.page-link:hover{
  color: #fff;
  background-color: #ce7852;
  border-color: #ce7852;
}
.page-link:focus{
  box-shadow: 0 0 0 0.2rem rgb(206, 120, 82, 25%);
}
.auth-button{
  display: inline-block;
  color: #fff;
  text-align: center;
  vertical-align: middle;
  background-color: transparent;
  border: 1px solid #ce7852;
  padding: .375rem .75rem;
  border-radius: .25rem;
  transition: background-color .15s;
}
form .auth-button{
  color: #000;
}
a.auth-button{
  text-decoration: none;
}
.auth-button:hover{
  background-color: #ce7852;
  color: #fff;
}
.form-control:focus, .custom-select:focus{
  color: #495057;
  background-color: #fff;
  border-color: #ce7852;
  outline: 0;
  box-shadow: 0 0 0 0.2rem rgb(206, 120, 82, 25%);
}
input[type=checkbox]{
  accent-color: #ce7852;
}
.bg-dark{
  background-color: #031546!important;
}
.navbar-dark .navbar-nav .nav-link{
  color: #fff;
}
.navbar-dark .navbar-nav .active>.nav-link{
  color: #ce7852!important;
}
.navbar-dark .navbar-nav .nav-link:focus, .navbar-dark .navbar-nav .nav-link:hover,
.navbar-dark .navbar-brand:hover, .navbar-dark .navbar-brand, .accent{
  color: #ce7852;
}
.navbar-dark .navbar-toggler {
  color: #ce7852;
  border-color: #ce7852;
}
button:focus {
  outline: 1px dotted;
  outline: 5px #ce7852;
}

```

```

}

.back-to-top {
  width: 40px;
  height: 40px;
  position: fixed;
  right: 15px;
  bottom: 15px;
  color: #ce7852;
  z-index: 5;
}

.back-to-top:hover {
  color: #ce7852;
  background-color: #cbd5e0;
  cursor: pointer;
}

.back-to-top i {
  font-size: 30px;
}

.hidden{
  display: none!important;
}

/*Loader CSS*/
@keyframes ldio-zftvpl388ch {
  0% { transform: translate(-50%,-50%) rotate(0deg); }
  100% { transform: translate(-50%,-50%) rotate(360deg); }
}

.ldio-zftvpl388ch div {
  position: absolute;
  width: 120px;
  height: 120px;
  border: 20px solid #ce7852;
  border-top-color: transparent;
  border-radius: 50%;
}

.ldio-zftvpl388ch div {
  animation: ldio-zftvpl388ch 1s linear infinite;
  top: 100px;
  left: 100px
}

.loadingio-spinner-rolling-5dqev8yz7mg {
  width: 200px;
  height: 200px;
  display: inline-block;
  overflow: hidden;
  background: #ffffff;
}

.ldio-zftvpl388ch {
  width: 100%;
  height: 100%;
  position: relative;
  transform: translateZ(0) scale(1);
  backface-visibility: hidden;
  transform-origin: 0 0; /* see note above */
}

.ldio-zftvpl388ch div { box-sizing: content-box; }

```

3.10 JavaScript

3.10.1 main.js

```
$(document).ready(function() {
  if($('#products').length) {
    filterProizvoda();
  }
  correctTotalPrice()

  $(document).on("click", ".addToCart", addToCart);

  $(document).on("change", ".genre", filterProizvoda);
  $('#search').keyup(filterProizvoda);
  $('#sort').on("change", filterProizvoda);

  $(document).on('change', '.quantity', changeQuantity);
  $(document).on('click', '.remove', removeFromCart);

  $(document).on('click', '.page-link', getPage);
})
//Ajax funkcija
function fetchData(url, method, data) {
  return new Promise((resolve, reject) => {
    $.ajax({
      url: url,
      method: method,
      dataType: "json",
      data: data,
      success: resolve,
      error: reject
    })
  })
}
//Funkcija za ispis html-a za proizvode
function ispisProizvoda(proizvodi) {
  let html = "";
  if(proizvodi.length > 0) {
    for(proizvod of proizvodi) {
      html += `

93


```

```

        </div>
    </div>`;
    }
}
else{
    html=`<div class="container not-found m-5 alert alert-danger"><p>There are no
products that match the criteria.</p></div>`
}
$('#products').html(html);
}

//Funkcije za prikaz, filter i paginaciju proizvoda
async function filterProizvoda(){
    let searchVal = $('#search').val().toLowerCase();
    let sortVal = $('#sort').val();
    let izabraniZanrovi=[];
    $('#genre:checked').each(function(el){
        izabraniZanrovi.push(parseInt($(this).val()));
    })
    window.filterObj = {
        search: searchVal,
        sort: sortVal,
        genres:izabraniZanrovi
    }
    try{
        $('#products .product, #products .not-found').remove();
        $('#loader').show();
        $('#pagination').hide();

        let books = await fetchData('/books/json', 'get', filterObj);

        $('#loader').hide();
        $('#count').html("Books (" +books.total+")")

        ispisProizvoda(books.data);
        printPagination(books.last_page);
    }catch(ex){
        console.error(ex);
    }
}

function printPagination(page,activePage = 1){
    let html = '';
    for(i =1;i<=page;i++){
        html += `<li class="page-item ${activePage == i ? 'active' : ''}"><a
class="page-link" href="#!" data-page="${i}">${i}</a></li>`;
    }
    $('#.pagination').html(html);
    $('#pagination').show();
}

async function getPage(){
    let page = $(this).data('page');
    let dataObj = filterObj;
    dataObj.page = page;
    try{
        $('#html').animate({
            scrollTop: $('#nav').offset().top
        }, 300);

        $('#products .product, #products .not-found').remove();
        $('#loader').show();
        $('#pagination').hide();

        let books = await fetchData('/books/json','get',dataObj);
    }
}

```

```

$("#loader").hide();
$('#count').html("Books (" + books.total + ")")

ispisProizvoda(books.data);
printPagination(books.last_page, books.current_page);
}catch(ex){
    console.error(ex);
}
}

//Funkcija za ispis poruke da je proizvod dodat u korpu
function poruka(tekst){
    switch(tekst){
        case 'uspeh':
            toastr.success('Added to cart!')
            break;
        case 'adminje':
            toastr.warning('Admin can not add to cart.');
```

break;

```

        case 'gostje':
            toastr.warning('You have to login first.')
```

break;

```

        case 'greska':
            toastr.warning('There has been an error processing your request.')
```

break;

```

    }
}

//Funkcije za obradu forme
function ukloniGreske(){
    $('#greska').hide();
    $('#uspeh').hide();
}

function greskaForme(elementForme, poruka){
    $(elementForme).next().text(poruka).fadeIn();
}

var izrazMejl = /^[a-z]{3,15})(([\.\]?[-]?[_]?[a-z]{3,20})*([\d]{1,3})*)([a-z]{3,20})(\.[a-z]{2,3})+$/;
var izrazImePrezime =
/^\p{Uppercase_Letter}\p{Letter}{1,14}(\s\p{Uppercase_Letter}\p{Letter}{1,14}){1,3}$/u;

$(document.regForm).on("submit", function(event){
    event.preventDefault();
    ukloniGreske();
    let forma=document.regForm;
    let greska=false;

    if(!izrazImePrezime.test(forma.tbName.value)){
        greskaForme(forma.tbName, "First and last name must start with capital letter and have at least 2 characters.")
        greska = true;
    }

    if(!izrazMejl.test(forma.tbEmail.value)){
        greskaForme(forma.tbEmail, "Please enter a valid email(example: yourname@gmail.com).")
        greska = true;
    }

    if(forma.tbPass.value.length < 7) {
        greskaForme(forma.tbPass, "Password must have at least 7 characters.");
        greska = true;
    }
}

```

```

    if(!greska){
        $('#btnReg').attr('type','hidden');
        forma.submit();
    }
})

$(document.logForm).on("submit",function(event){
    event.preventDefault();
    ukloniGreske();
    let forma = document.logForm;
    let greska = false;
    console.log("desilo se")
    if(!izrazMejl.test(forma.tbEmail.value)){
        greskaForme(forma.tbEmail, "Please enter a valid email(example:
yourname@gmail.com).")
        greska = true;
    }
    if(!forma.tbEmail.value){
        greskaForme(forma.tbEmail,"Email field can't be empty.");
        greska=true;
    }
    if(!forma.tbPass.value){
        greskaForme(forma.tbPass,"Password field can't be empty.");
        greska=true
    }

    if(!greska){
        $('#btnLogin').attr('type','hidden');
        forma.submit();
    }
})

$(document.conForm).on("submit",function(event){
    event.preventDefault();
    ukloniGreske();
    let forma=document.conForm;
    let greska=false;

    if(!izrazImePrezime.test(forma.tbName.value)){
        greskaForme(forma.tbName,"First and last name must start with capital letter
and have at least 2 characters.")
        greska = true;
    }

    if(!izrazMejl.test(forma.tbEmail.value)){
        greskaForme(forma.tbEmail, "Please enter a valid email(example:
yourname@gmail.com).")
        greska = true;
    }
    if(!forma.tbEmail.value){
        greskaForme(forma.tbEmail,"Email field can't be empty.");
        greska=true;
    }
    if(forma.userMessage.value.length == "" || forma.userMessage.value.length > 200) {
        greskaForme(forma.userMessage, "Please enter a message no longer than 200
characters.");
        greska = true;
    }

    if(!greska){
        $('#btnSubmitContact').attr('type','hidden');
        forma.submit();
    }
})

```



```

//Scroll to top dugme
const showOnPx = 200;
const backToTopButton = document.querySelector(".back-to-top");

const scrollContainer = () => {
    return document.documentElement || document.body;
};

const goToTop = () => {
    document.body.scrollToView({
        behavior: "smooth"
    });
};

document.addEventListener("scroll", () => {

    if (scrollContainer().scrollTop > showOnPx) {
        backToTopButton.classList.remove("hidden");
    } else {
        backToTopButton.classList.add("hidden");
    }
});

backToTopButton.addEventListener("click", goToTop);

//Funkcije za rad sa korpom
function addToCart(){
    let id=$(this).data('id');
    $.ajax({
        method: "POST",
        url: "/cart",
        data: {
            bookId : id
        },
        success: function(data) {
            poruka(data)
        },
        error: function(xhr) {
            console.error(xhr);
        }
    })
}

function changeQuantity() {
    let inputVal = $(this).val();
    let price = $(this).data('price');
    let bookId = $(this).data('id');
    console.log(inputVal)
    if(!inputVal || inputVal < 1) {
        toastr.warning('Only numbers equal to or higher than 1 are allowed in the quantity field.')
        $(this).val(1);
    } {
        $("#span_price_" + bookId).html($(this).val() * price);
        correctTotalPrice();

        $.ajax({
            method: "POST",
            url: "/cart/adjustQuantity",
            data: {
                bookId : bookId,
                quantity : inputVal
            }
        })
    }
}

```

```

        },
        success: function() {
            toastr.success("Adjustment successfull.")
        },
        error: function(xhr, status, error) {
            toastr.error("There was an error processing your request.")
        }
    })
}
}

function correctTotalPrice() {
    var allPrices = document.querySelectorAll(".product_price")

    var totalPrice = 0;

    for(let priceSpan of allPrices) {
        totalPrice += Number(priceSpan.innerHTML)
    }

    $("#totalPrice").html(totalPrice)
}

function removeFromCart() {
    let bookId = $(this).data('id');
    $.ajax({
        method: "DELETE",
        url: "/cart/" + bookId,
        success: function() {
            toastr.success("Item successfully removed.")

            let cartItemDiv = $("#cart_item_" + bookId)

            cartItemDiv.remove()
            if($('.cart-item').length == 0){
                location.reload();
            }
            $("#count").html($('.cart-item').length+" items");
            correctTotalPrice()
        },
        error: function(xhr, status, error) {
            toastr.error("There was an error processing your request.")
        }
    })
}
}

```