

READILY

Dokumentacija

Nikola Samardžić 54/21

Praktikum iz PHP-a

Sadržaj

Funkcionalnosti	1
Organizacija	4
Dizajn baze podataka:	4
Mapa sajta:	4
Opis funkcionalnosti na stranici:	9
Home	9
Signup	14
Login	18
Writer	19
Shop	20
Checkout	21
Book	26
Admin	31
Admin-users	32
Admin-categories	33
Admin-publishers	35
Admin-delivery-options	36
Admin-survey	37
Admin-messages	38
Admin-orders	39
Admin-visitors	40
Admin-logged-in-users	40
Writer-books	41
Insert-book	42
Update-book	44
404	46
Kodovi	47
.env	47
Config.php	48
Connection.php	49
Activate-book.php	50
Add-book.php	52
Book-rating.php	55
Delete-book.php	57
Models > book > functions.php	59

Get-book-active-categories.php	71
Get-suggested-books.php	72
Insert-comment.php	74
Update-book.php	77
Cart-logic.php	81
Models > cart > functions.php	84
Activate-category.php	88
Add-update-categories.php	90
Delete-category.php	93
Get-active-categories.php	95
Models > category > functions.php	96
Activate-delivery-type.php	103
Add-update-delivery-type.php	105
Delete-delivery-type.php	107
Models > delivery > functions.php	109
Models > footer > functions.php	111
Models > global > fucntions.php	113
Models > header > functions.php	114
Models > home > functions.php	115
Models > login > functions.php	116
login.php	117
Models > message > functions.php	120
Send-message.php	121
Models > order > functions.php	123
order-logic.php	125
Activate-publisher.php	129
Add-update-publisher.php	131
Delete-publisher.php	133
Models > publisher > functions.php	135
shop.php	137
Activate-account.php	139
Models > signup > functions.php	140
signup.php	143
Activate-survey.php	148
Delete-survey.php	150
Models > survey > functions.php	152
Insert-survey-webisite-experience-data.php	155

Activate-user.php	157
Ban-user.php	159
Delete-user.php	161
Models > user > functions.php	163
Unban-user.php	172
Update-user-info.php	174
Models > writer > functions.php	180
Footer.php	184
Head.php	185
Header.php	186
404.php	188
Admin-categories.php	189
Admin-delivery-options.php	193
Admin-logged-id-users.php	196
Admin-messages.php	198
Admin-orders.php	200
Admin-publishers.php	202
Admin-survey.php	205
Admin-users.php	208
Admin-visitors.php	212
Admin.php	214
Author.php	217
Book.php	218
Checkout.php	226
Home.php	231
Insert-book.php	237
Login.php	240
Logout.php	241
Shop.php	242
Signup.php	244
Update-book.php	248
User-profile.php	252
Writer-books.php	259
Writer.php	262
Index.php	265
Main.js	266
_animations.scss	360

_bg-colors.scss	361
_reset.scss	363
_responsive.scss	364
_variables.scss	366
style.scss	367

Uvod

Korišćene tehnologije:

- HTML
- CSS
- SCSS
- JavaScript
- PHP
- jQuery
- Ajax
- MySQL
- PhpMyAdmin

Za izradu sajta nije korišćen templejt

Link ka github repo:

<https://github.com/NikolaSamardzic/Readilyphp>

Funkcionalnosti

- **Dinamičko ispisivanje:** linkova kod menija u header-u i footer-u, artikala knjiga, kategorija i pisaca, informacija o knjigama, komentara i njihovih slika, sadržaj svake strane koja se nalazi u admin panelu, dinamički meni u vidu kategorija i podkategorija za potrebu filtriranja na strani shop.php, artikala koji su u korpi, opcije kod ankete

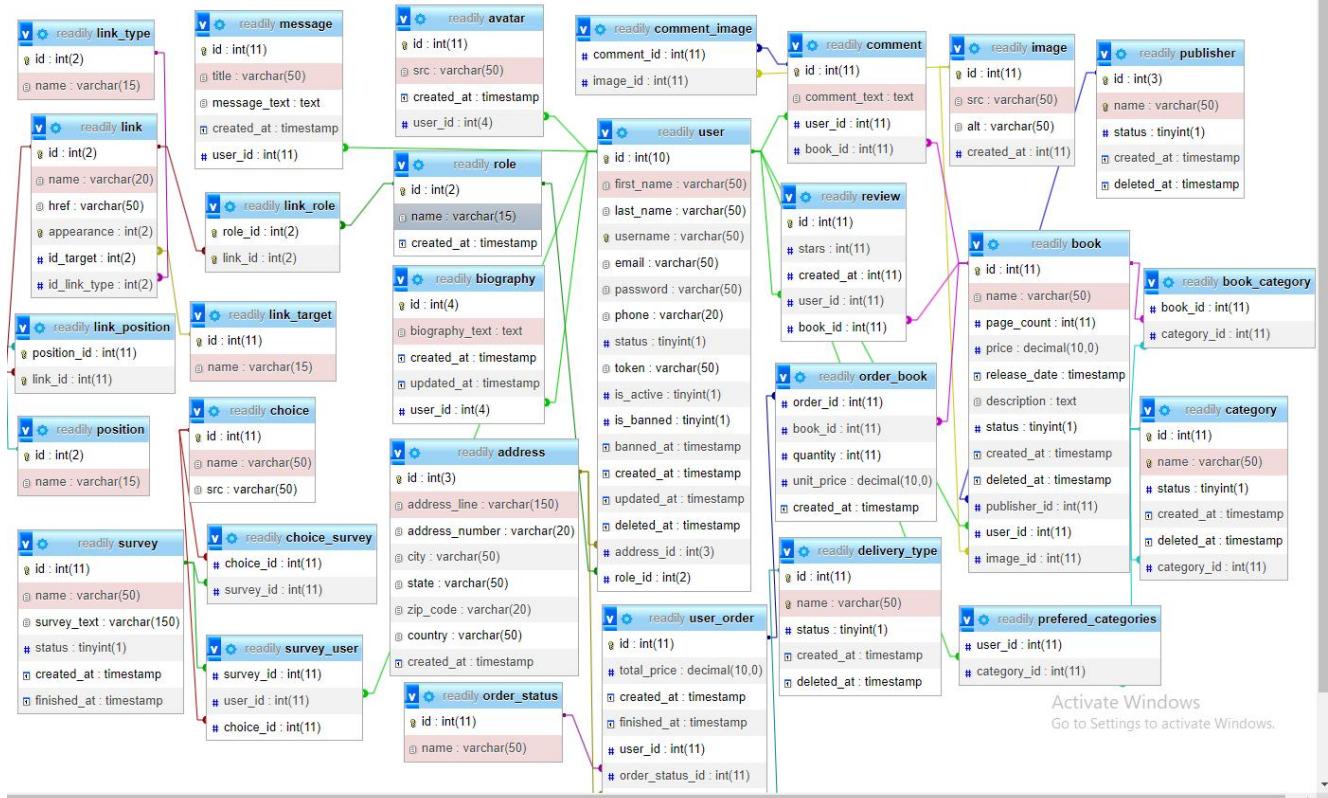
- **Registracija** - korisnik ima mogućnost da izostavi informacije o adresi i profilnoj slici. Ukoliko izostavi sliku, na delovima sajta gde se ona traži, biće korišćena unapred definisana slika default.jpg (ovo se vrši u php-u kodom a ne u bazi). Ukoliko unese adresu koja već postoji u sistemu (ukoliko još neko u njegovoj porodici ima nalog), biće mu dodeljena referenca te iste adrese a ukoliko odluči da ukloni informacije o adresi koja je referencirana od više korisnika, samo će kod njega biti ulonjena a adresa ostaje u sistemu. Lozinka se štiti dodatnim ‘saltingom’ gde se na postojeću vrednost dodje određeni broj karaktera i tek onda dolazi do enkripcije

- **Logovanje**
- **Aktivacija naloga putem mejla**
- **Autorizacija** - na osnovu uloge koja mu se dodeli prilikom kreiranja naloga, korisniku će biti vidljivi odredjeni delovi sajta i biće mu omogućene određene funkcionalnosti
- **Klijentska i serverska provera podataka**
- **Zaštita od pristupa stranicama putem linka**
- **Mogućnost editovanja naloga i informacija o knjigama**
- **Filtriranje i sortiranje**
- **Kupovina**
- **Mogućnost komentarisanja i ocenjivanja knjige**
- **Anketiranje korisnika**
- **Kontakt forma za korisnika**
- **Dodavanje slika kod komentara**
- **Paginacija proizvoda na shop strani**

- **Prikaz obaveštenja o uspešnosti i neuspešnosti izvršavanja određenih funkcionalnosti na serveru**
- **Korišćenje AJAXA** - sve forme se šalju putem AJAXA a ukoliko server treba da prosledi određene infomacije nazad na front, u success metodu će biti pozvane funkcije za ispis
- **Korišćenje kolačića** - za određene funkcionalnosti poput, da li je korisnik ulogovan, informacije o nalogu, za potrebu dodavanje artikala u korpu, light/dark mode
- **Kreiranje thumbnail slike** - prilikom uploada slike knjige pomoću php-a se kreiraju 2 slike različitih dimenzija i kvaliteta. Manja se koristi prilikom prikaza više knjiga dok se veća koristi prilikom prikaza pojedinačne knjige
- **Log fajl** - vođenje evidencije o posetama na stranicama i evidencija o korisnicima koji su se ulogovali u poslednja 24 časa

Organizacija

Dizajn baze podataka:



Mapa sajta:

```
<?xml version="1.0" encoding="UTF-8"?>
<urlset
    xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.sitemaps.org/schemas/sitemap/0.
9
    http://www.sitemaps.org/schemas/sitemap/0.9/sitemap.xsd">
```

```

<url>
    <loc>https://readilyphp.000webhostapp.com/index.php?page=home</loc>
    <lastmod>2023-5-12</lastmod>
    <changefreq>daily</changefreq>
    <priority>1</priority>
</url>

<url>
    <loc>https://readilyphp.000webhostapp.com/index.php?page=shop</loc>
    <lastmod>2023-5-12</lastmod>
    <changefreq>daily</changefreq>
    <priority>0.95</priority>
</url>

<url>
    <loc>https://readilyphp.000webhostapp.com/index.php?page=book</loc>
    <lastmod>2023-5-12</lastmod>
    <changefreq>daily</changefreq>
    <priority>0.92</priority>
</url>

    <url>
        <loc>https://readilyphp.000webhostapp.com/index.php?page=author</loc>
        <lastmod>2023-5-12</lastmod>
        <changefreq>daily</changefreq>
        <priority>0.91</priority>
    </url>

    <url>
        <loc>https://readilyphp.000webhostapp.com/index.php?page=checkout</loc>
        <lastmod>2023-5-12</lastmod>
        <changefreq>daily</changefreq>
        <priority>0.91</priority>
    </url>

    <url>

<?xml version="1.0" encoding="UTF-8"?>
<urlset
    xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.sitemaps.org/schemas/sitemap/0.
9
```

```
    http://www.sitemaps.org/schemas/sitemap/0.9/sitemap.xsd">

<url>
    <loc>https://readilyphp.000webhostapp.com/index.php?page=home</loc>
        <lastmod>2023-5-12</lastmod>
        <changefreq>daily</changefreq>
        <priority>1</priority>
</url>

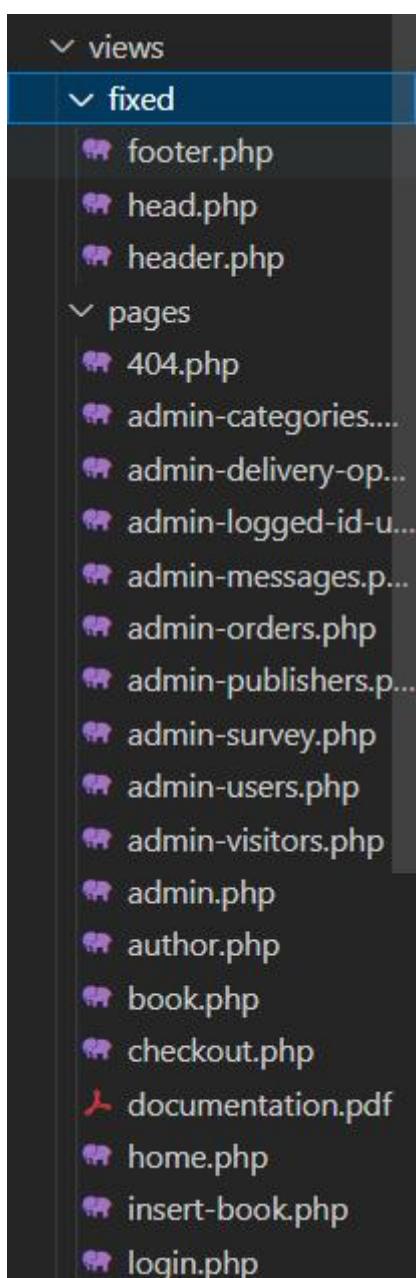
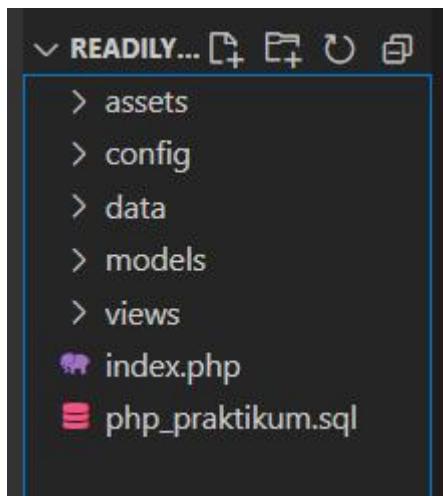
<url>
    <loc>https://readilyphp.000webhostapp.com/index.php?page=shop</loc>
        <lastmod>2023-5-12</lastmod>
        <changefreq>daily</changefreq>
        <priority>0.95</priority>
</url>

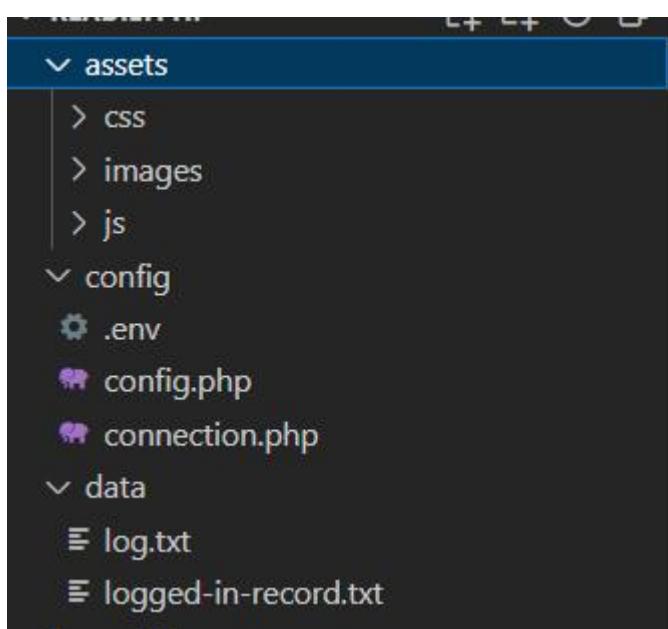
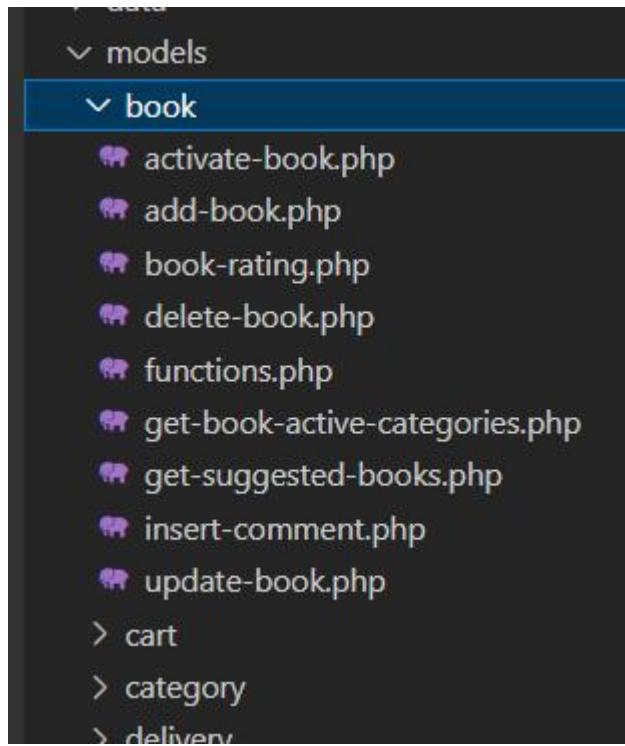
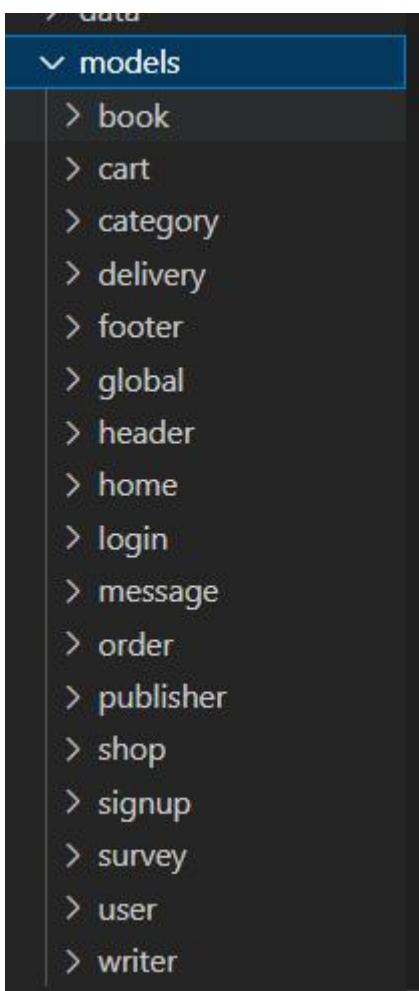
<url>
    <loc>https://readilyphp.000webhostapp.com/index.php?page=book</loc>
        <lastmod>2023-5-12</lastmod>
        <changefreq>daily</changefreq>
        <priority>0.92</priority>
</url>

<url>
    <loc>https://readilyphp.000webhostapp.com/index.php?page=author</loc>
        <lastmod>2023-5-12</lastmod>
        <changefreq>daily</changefreq>
        <priority>0.91</priority>
</url>

<url>
    <loc>https://readilyphp.000webhostapp.com/index.php?page=checkout</loc>
        <lastmod>2023-5-12</lastmod>
        <changefreq>daily</changefreq>
        <priority>0.91</priority>
</url>

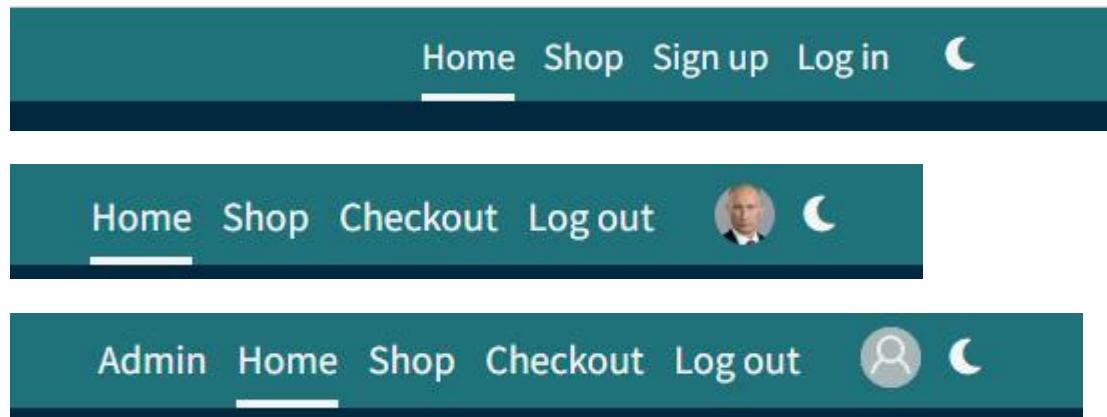
<url>
```





Opis funkcionalnosti na stranici:

Home



U zavisnosti od uloge, korisniku će se prikazati linkovi namenjeni toj ulozi

A screenshot of the Readily website. The top navigation bar includes 'Readily' logo, 'Admin' (underlined), 'Home', 'Shop', 'Checkout', 'Logout', a magnifying glass icon, and a moon icon. The main content area features a section titled 'What is Readily?' with a paragraph about the company's passion for books. Below this is a product card for 'Harry Potter and the Prisoner of Azkaban' by J.K. Rowling. The card shows a 5-star rating (5/5, 51052 ratings), a 'About this book' summary, and a large image of the book cover. The price is listed as '\$15.66' with a shopping cart icon. At the bottom right of the page, there are links for 'Activate Windows', 'Go to Settings to activate Windows.', and 'Powered by 000webhost'.

Readily

Having become classics of our time, the Harry Potter eBooks never fail to bring comfort and escapism. With their message of hope, belonging and the enduring power of truth and love, the story of the Boy Who Lived continues to delight generations of new readers.

Activate Windows
Go to Settings to activate Windows.

Admin Home Shop Checkout Log out

Whatever you're into, we've got something for you

Discover something new every day
At Readily, we're all about discovering something new. With our vast selection of titles and new releases added regularly, there's always something fresh and exciting to explore. Join us today and start your journey of discovery.

Originals you won't find elsewhere
At Readily, we pride ourselves on offering a selection of books that you won't find anywhere else. From rare editions to limited releases, we're passionate about bringing you unique and original titles that will inspire and delight you. Join us today and discover the joy of owning a one-of-a-kind book.

Help finding your next great read
At Readily, we're here to help you find your next favorite book. Whether you're looking for a specific genre, author, or just need some inspiration, our knowledgeable staff are on hand to offer recommendations and advice. With our user-friendly website and personalized service, discovering your next great read has never been easier. Join us today and let us help you find your next literary obsession.

Go to Settings to activate Windows.

Powered by 000webhost

Ukoliko novi korisnik dođe do ovog dela sajta, prikazaće mu se sekcija na kojoj će selektovati kategorije koje mu se najviše dopadaju i na osnovu tih kategorija će mu putem AJAX-a biti prosleđene knjige koje spadaju u te kategorije. Svakim sledećim ulaskom na stranicu ta sekcija mu se neće prikazati a knjige koje najviše odgovaraju tom izboru biće prikazane Server Side Renderingom

Select 3 categories that match your interests

Mystery	Travel	Self Improvement
Cooking	Technology And Science	Children's Books
Romance	History	Business And Economics
Fiction		

Activate Windows
Go to Settings to activate Windows.

Knjige koje će mu se prikazati nakon selekcije:

Books Recommended For You

Fairy Tale
Stephen King
★★★★★ 0 ratings

See Me
Nicholas Sparks
★★★★★ 0 ratings

Harry Potter and the Order of the Phoenix
J.K. Rowling
★★★★★ 0 ratings

Death on the Nile
Agatha Christie
★★★★★ 0 ratings

The Adventures Of Sherlock Holmes
Sir Arthur Conan Doyle
★★★★★ 0 ratings

Nightwork
Nora Roberts
★★★★★ 0 ratings

Readily
Bestselling Books

Harry Potter and the Prisoner of Azkaban
J.K. Rowling
★★★★★ 4/5

Unshakeable: Your Financial Freedom Playbook
Tony Robbins
★★★★★ 4/5

The Cat in the Hat
Dr. Seuss
★★★★★ 5/5

Charlie and the Great Glass Elevator
Roald Dahl
★★★★★ 4/5

How the Grinch Stole Christmas!
Dr. Seuss
★★★★★ 4/5

Harry Potter and the Order of the Phoenix
J.K. Rowling
★★★★★ 0 ratings

Popular Categories

Mystery
And Then There Were None by Agatha Christie

Travel
Notes From a Small Island by Bill Bryson

Self Improvement
How to Win Friends and Influence People by Dale Carnegie

Cooking
Mastering the Art of French Cooking by Martha Stewart

Technology And Science
Hackers by Eric S. Raymond

Anketa koja će se prikazati korisniku

Please rate your experience on our website and help us improve!



Terrible



Poor



Average



Good



Excellent

Odgovor koji će dobiti nakon što je popuni

Please rate your experience on our website and help us improve!

Survey information saved successfully. Thank you for your participation!

Forma za kontakt

Get in Touch with Us

Subject

Message

Send

ANALIZA I VJEZBE

Odgovor sa servera nakon što je poruka uspešno upisana u bazu

Get in Touch with Us

Subject

Message

Thank you! Your message has been sent successfully

Send



Signup

Personal info

upload avatar
Choose File download (1).jpg

First Name: Vladimir Last Name: Putin

Username: Vlada123 Password:
Email: pp5104133@gmail.com

Activate Windows
Go to Settings to activate Windows.

Powered by 000webhost

Svaki podatak se proverava kako na klijentu, tako i na serveru.

Username mora biti jedinstven a ukoliko nije, korisnik će biti obavešten porukom sa servera

First Name: Vladimir Last Name: Putin

Username: Vlada123 Password:
Email: pp5104133@gmail.com

Phone: 1234567 Role: Customer

Address info
(optional)

Address line: Test

Number:

Activate Windows
Go to Settings to activate Windows.

Powered by 000webhost

Address info

(optional)

Address line

Number

City

State

Zip-code

Country

Create AccountActivate Windows
Go to Settings to activate Windows.Powered by  000webhost

Informacije o adresi nisu neophodne, a ukoliko adresa postoji u sistemu, korisniku će biti dodeljena referenca na nju.

Poruka o grešci za unique username:

City

State

Zip-code

Country

Server error: SQLSTATE[23000]: Integrity constraint violation: 1062 Duplicate entry 'Admin' for key 'username'

Create Account

Mejl za aktivaciju nalogu

The screenshot shows an email interface with a dark background. At the top, there's a header with a gray octagonal profile picture containing a red question mark, the name "Readily" in white, the time "20:42" in light gray, and a dropdown menu "коме: мени". To the right are a back arrow and a three-dot menu icon.

Сакриј цитирани текст

Activate Your Account

Hi,

Thanks for creating an account with us! Please activate your account by clicking on the following link:

[Activate Your Account](#)

If you have any questions or concerns, don't hesitate to contact us.

Best regards,
Readily

Poruke o greškama vezane za polja:

Address line

Please enter a valid address line.

Number

City

Incorrect format (ex. Los Angeles)

State

Zip-code

Country

Poruka o uspešnosti kreiranja naloga:

City

State

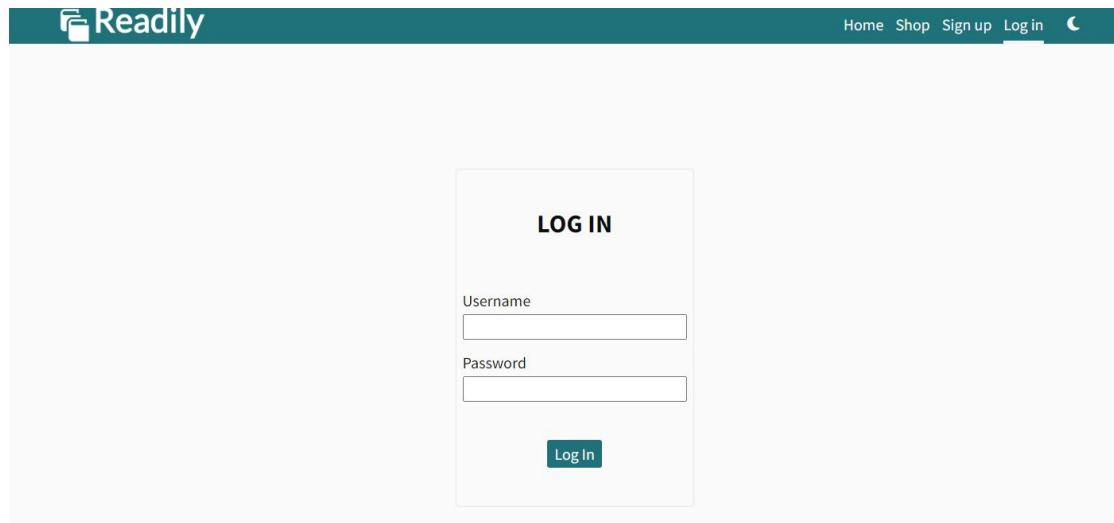
Zip-code

Country

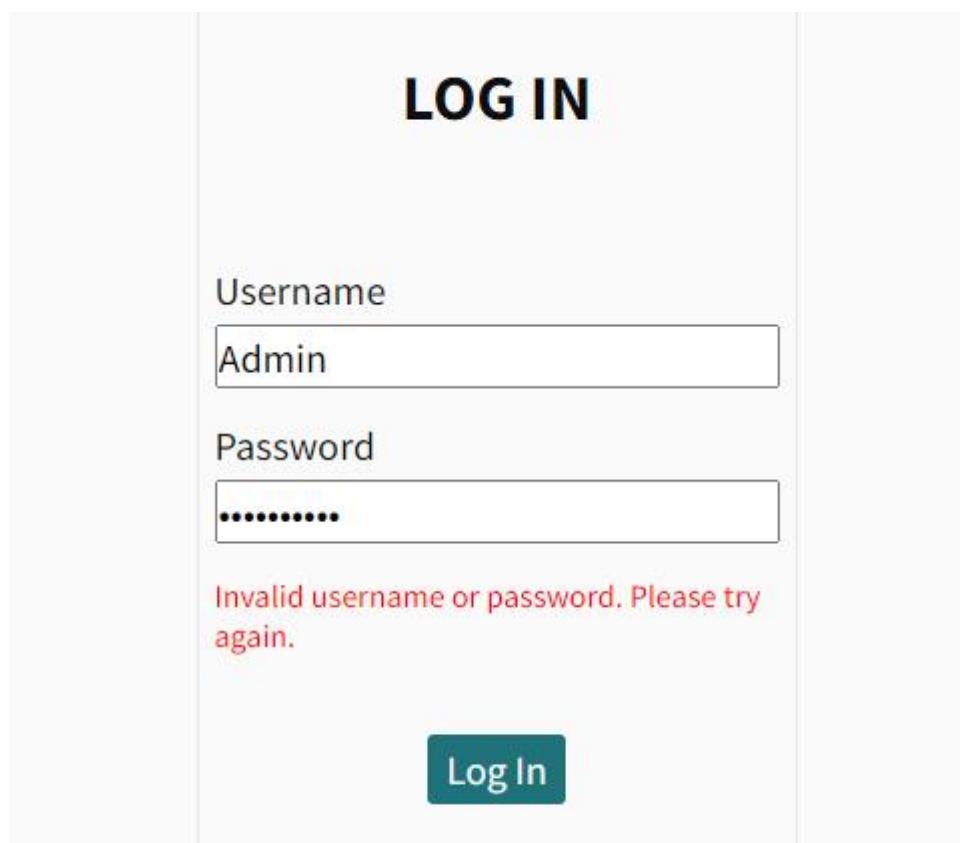
Hi! Your account has been created successfully. Please check your email for activation instructions. Thank you!

[Create Account](#)

Login



Ukoliko dođe do greške prilikom logovanja, korisnik će biti obavešten



Writer

Sadrži biografiju i sliku o piscu, knjige koje je izdao, kategorije u kojima mu se nalaze knjige, linkove ka ostalim piscima.

 **Readily**

Home Shop Checkout Log out 

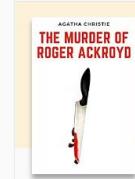
 **Agatha Christie**

Born in Torquay in 1890, Agatha Christie began writing during the First World War and wrote over 100 novels, plays and short story collections. She was still writing to great acclaim until her death, and her books have now sold over a billion copies in English and another billion in over 100 foreign languages. Yet Agatha Christie was always a very private person, and though Hercule Poirot and Miss Marple became household names, the Queen of Crime was a complete enigma to all but her closest friends. [Show Less](#)

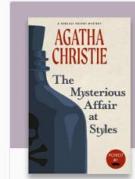
Books by J.K. Rowling

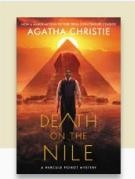

Murder on the Orient Express
Agatha Christie
 0 ratings 


And Then There Were None
Agatha Christie
 0 ratings 


The Murder of Roger Ackroyd
Agatha Christie
 0 ratings 


Midwinter Murder
Agatha Christie
 0 ratings 


The Mysterious Affair at Styles
Agatha Christie
 0 ratings 


Death on the Nile
Agatha Christie
 0 ratings 

 Go  activate Windows.
Powered by 

 **Readily**

Home Shop Checkout Log out 

Related Categories

 Mystery

Check Out Other Authors

 **Agatha Christie**
10 titles

 **Sir Arthur Conan Doyle**
5 titles

 **Bill Bryson**
10 titles

 **Paul Theroux**
5 titles

 **Dale Carnegie**
10 titles

 **Tony Robbins**
5 titles

 Go  activate Windows.
Powered by 

Author Documentation
RSS
Sitemap

Facebook
Instagram
Twitter

Home
Shop
Checkout
Go to Settings to activate Windows.
Powered by 

Shop

Nalaze se izlistane sve knjige koje imaju parametre u skladu sa parametrima koje korisnik traži.

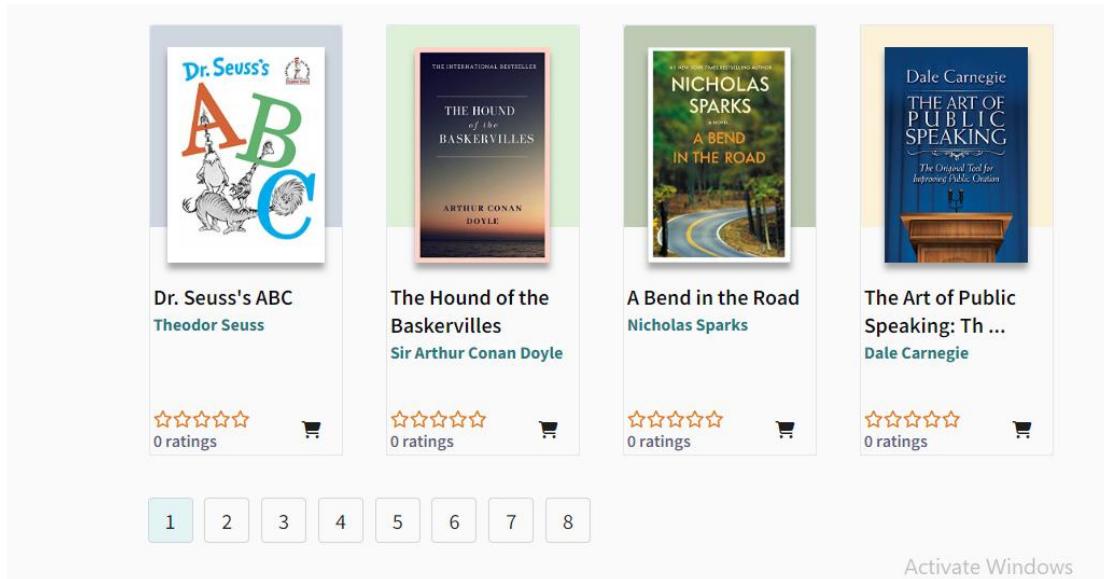
Podmeni realizovan u vidu kategorija po kojima se filtriraju knjige.

The screenshot shows the Readily bookshop interface. At the top, there is a navigation bar with links for Home, Shop, Checkout, Log out, and user profile icons. On the left, there is a sidebar with search and filter options. The search bar has a placeholder 'Search' and a magnifying glass icon. Below it are 'Price' filters ('Min' and 'Max') and a 'Categories' dropdown menu containing items like Mystery, Travel, Self Improvement, etc. Under 'Sort', there is a dropdown set to 'Most Popular'. The main area displays a grid of nine book covers. Each cover includes the title, author, rating (e.g., 4/5), and a shopping cart icon. The books shown are: Harry Potter and the Prisoner of Azkaban by J.K. Rowling, Unshakeable: Your Financial Freedom Plan by Tony Robbins, The Cat in the Hat by Dr. Seuss, Charlie and the Great Glass Elevator by Roald Dahl, How the Grinch Stole Christmas! by Dr. Seuss, Agatha Christie's Midwinter Murder, Stephen King's The Institute, and One Fish, Two Fish, Red Fish, Blue Fish by Dr. Seuss. A small note at the bottom right says 'Powered by 000webhost'.

Ukoliko nema ni jedne knjige za zadati kriterijum, ispisaće se poruka:

The screenshot shows the Readily bookshop interface again. The navigation bar and sidebar are identical to the previous screenshot. In the main area, there is a message: 'We're sorry, but we couldn't find any books that match your interests.' This message is displayed in red text. The rest of the interface is identical to the one above, showing the search bar, price filters, categories, and a grid of book covers.

Filtriranje, sortiranje i straničenje se vrši na serveru



Checkout

Funkcionalnost kupovine:

1. Kada korisnik klikne na ikonicu korpe, update-ovaće se vrednost kolačića 'cart'
2. Nakon update-ta, šalje se AJAX zahtev ka serveru
3. Na serveru se dohvata vrednost kolačića i smeštaju se u promenljivu, u drugu promenljivu se smeštaju artikli koji se trenutno nalaza u bazi.
4. Prave se 3 nove promenljive koje će sadržati nizove objekata prilikom preseka ova dva niza.

5. U jednoj će se nalaziti objekti sa informacijama koje trebaju da se unesu u bazu u drugoj informacije o redovima koji trebaju da se izbrišu iz baze a u trećoj redovi koji trebaju da se update-ju (količina knjiga)

Na ovoj stranici se prikazuju informacije o korisničkoj korpi i knjigama koje se u njoj nalaze.

Korisnik može da menja njihovu količinu i uklanja iz korpe a u pozadini će se odvijati gore navedena funkcionalnost.

AJAX-om će se pozvati f-ja za ispis proizvoda i update ukupne cene

Ukoliko nema proizvoda u korpi, biće mu prikazana poruka:

Image	Title	Unit Price	Quantity	Price	Remove
The Cart Is Empty					
					\$0.00

Ukoliko je korisnik uneo informacije o svojoj adresi prilikom kreiranja naloga, te informacije će mu biti unete u polja za adresu, dok ukoliko nije, moraće ručno da ih unese.

Address info

Address line

Number <input type="text" value="12"/>	Delivery option <input type="text" value="Standard"/>
City <input type="text" value="Test"/>	State <input type="text" value="Test"/>
Zip-code <input type="text" value="123412"/>	Country <input type="text" value="Test"/>

[Activate Wi-Fi](#)
[Go to Settings](#)

Nakon što potvrди kupovinu, korisnik će biti obavešten o njenoj uspešnosti/neuspešnosti a podaci i korpa će biti ispraznjeni

City <input type="text"/>	State <input type="text"/>
Zip-code <input type="text"/>	Country <input type="text"/>

Thank you for your order! Your purchase has been saved and will be processed soon.

User-profile

Nalaza se informacije o korisniku koje može da edituje ili sam korisnik ili admin.

U kolačiću se nalaze informacije o korisniku zbog eventualne situacije gde korisnik se odluči da edituje informacije, izmeni ih ili obriše, odustane od njihovog menjanja i ponovo se odluči na njihovu izmenu, pošto ova stranica sadrži dve sekcije koje se pojedinačno prikazuju, kolačićima je uspostavljena funkcionalnost ponovnog upisa informacija o korisniku u polja kako ona ovoga puta ne bi bila prazna.

Nakon što potvrди unos podataka, korisnik će biti vraćen na početak stranice ukoliko je usešna promena odnosno biće obavešten o njenoj neuspešnosti u vidu poruke.

Korisnik takođe može obrisati svoj nalog i u tom slučaju biće odjavljen sa naloga i redirektovan na stranicu index.php a u bazi će biti odraćen soft delete

Personal info



First Name

Vladimir

Last Name

Putin

Username

Vlada123

Email

pp5104133@gmail.com

Phone

1234567

Role

Customer

Act
Go to

Address info

Address line

Test

Number

12

City

Test

State

Test

Zip-code

123412

Country

Test

Delete

Update

Acti

Book

Na ovoj stranici se prikazuju informacije o selektovanoj knjizi:

- Naslov
- Link ka stranici autora
- Broj ocena
- Broj komentara
- Prosečna ocena
- Rezime o knjizi
- Cena
- Kategorije u kojima se nalazi
- Broj strana
- Izdavač
- Datum izdavanja
- Thumbnail slike proizvoda i slika većih dimenzija za selektovanu knjigu

Takođe se nalaze dve sekcije, jedna sa knjigama koje imaju iste kategorije kao zadata knjiga i druga sa svim ostalim knjigama koje su izdate od strane istog pisca

Harry Potter and the Prisoner of Azkaban

By [Joanne Rowling](#)

 4/5 (7 ratings)

About this book

'Welcome to the Knight Bus, emergency transport for the stranded witch or wizard. Just stick out your wand hand, step on board and we can take you anywhere you want to go.'

When the Knight Bus crashes through the darkness and screeches to a halt in front of him, it's the start of another far from ordinary year at Hogwarts for Harry Potter. Sirius Black, escaped mass-murderer and follower of Lord Voldemort, is on the run - and they say he is coming after Harry. In his first ever Divination class, Professor Trelawney sees an omen of death in Harry's tea leaves... But perhaps most terrifying of all are the Dementors patrolling the school grounds, with their soul-sucking kiss...

Having become classics of our time, the Harry Potter eBooks never fail to bring comfort and escapism. With their message of hope, belonging and the enduring power of truth and love, the story of the Boy Who Lived continues to delight

Having become classics of our time, the Harry Potter eBooks never fail to bring comfort and escapism. With their message of hope, belonging and the enduring power of truth and love, the story of the Boy Who Lived continues to delight generations of new readers.

Related Categories

Mystery



Children's Books



Fiction



Pages
180

Publisher
Pottermore Publishing

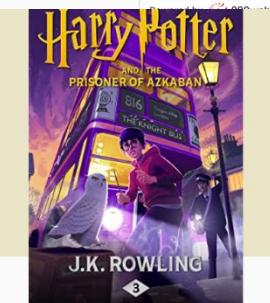
Release Date
March 20, 2015



\$15.00



Activate Windows
Go to Settings to activate Windows.



\$15.00

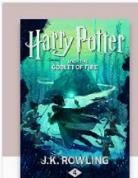


Related to Harry Potter and the Prisoner of Azkaban



Harry Potter and the Order of the Phoenix
Joanne Rowling

 0 ratings



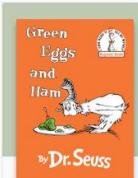
Harry Potter and the Goblet of Fire
Joanne Rowling

 0 ratings



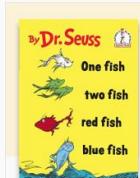
Gwendy's Final Task
Stephen King

 0 ratings



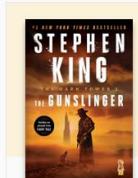
Green Eggs and Ham
Theodor Seuss

 0 ratings



One Fish Two Fish
Red Fish Blue Fish
Theodor Seuss

 2/5



The Dark Tower I:
The Gunslinger
Stephen King

 0 ratings



[Read more from Joanne Rowling](#)



Harry Potter and the Goblet of Fire
Joanne Rowling

★★★★★
0 ratings



Harry Potter and the Half-Blood Prince
Joanne Rowling

★★★★★
0 ratings



Harry Potter and the Prisoner of Azkaban
Joanne Rowling

★★★★★
4/5



Harry Potter and the Order of the Phoenix
Joanne Rowling

★★★★★
0 ratings



Harry Potter and the Sorcerer's Stone
Joanne Rowling

★★★★★
0 ratings



The Ink Black Heart
Joanne Rowling

★★★★★
0 ratings

Ukoliko je ulogovan, korisniku će se prikazati sekcija za ocenjivanje knjige i ostavljanje komentara

Rate this Book



Write a Review

Comment

There must be at least 5 words.

Images

Choose File No file chosen

A
Gc

Nakon što unese ocenu ili komentar, biće obavešten sa odgovarajućom porukom.



Your rating has been successfully saved.

Write a Review

Comment

Images

No file chosen

Your comment has been successfully uploaded.

Input polje za slike kod komntara se dinamički dodaje/uklanja putem JS na osnovu toga da li je korisnik dodao sliku ili uklonio sliku iz inputa, maksimalan broj slika je 3.

Takođe se prikazuju svi komentari i njihove slike ukoliko ih imaju



BookishBibliophile

I bought this for my daughter who is getting into Harry Potter now. I didn't want to spend a lot of money since I'm trying to buy all the books. I'm very satisfied with the seller, the book matched their description perfectly.

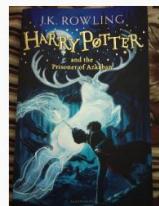
★★★☆☆ 3/5



BusinessBoss

★★★★★ 5/5

As always with Rowling's books, I loved Harry Potter and the Prisoner of Azkaban! JK Rowling's words have a curious habit of coming to life, and her characters are funny, and realistic. One of the greatest things about Harry Potter, is that they improve with each book, and you can clearly glimpse the clever, intricate plot Rowling has wove, with cleverly placed foreshadowing and seemingly innocent hints.



Activate Windows



CultureCrusader

★★★★★ 5/5

In this book, as Harry approaches adolescence the tone changes. And we get a more interesting, but also more complex and darker story.

Unlike the static characters in other books or series, Harry is getting older, and getting less stable emotionally. The book still appeals to both children and adults but it involves some horror creatures, Dementors and there is a lot of focus on Harry's parents' deaths.

The great thing for children is that the familiar, relatable characters are being brought to life and come across as very realistic. Harry's friendship with Ron and Hermione takes a different direction in this instalment. Hermione is being forced to make tough choices that impact Harry but Ron in particular. With who she is in constant disagreement. But the book also has an exceptionally strong set of secondary characters like Professor Lupin, the Defense Against the Dark Arts teacher, a great new character that hides a big hairy secret. Then there is Professor Trelawney for Divination studies. While Hagrid displays his love of all magical creatures. Professor Snape remains enigmatic so that you continue to question his true motives. Cedric Digory and Cho Chang, who play important roles in later books of the series are introduced as well



Activate Windows
Go to Settings to activate Windows.

Admin

Ovoj stranici može pristupiti samo admin i na njoj se nalaze linkovi ka ostalim stranicama koje imaju funkcionalnosti namenjene samo adminu.

Takođe, tim stranicama samo admin može pristupiti

The image shows a user interface for an administrative panel. It is organized into several sections:

- Tables**: A row of four database icons, each with a corresponding label:
 - Users
 - Categories
 - Publishers
 - Delivery Options
- Surveys**: A single survey icon labeled "Survey Result".
- Messages**: A single message icon labeled "User Messages".
- Statistics**: Three chart icons representing different metrics:
 - Orders
 - Visitors
 - Logged-in Users

Admin-users

Na ovoj stranici se nalaze tabele za aktivne, banovane, i izbrisane naloge.

Stanje naloga se može promeniti klikom na dugme Ban/Unban/Delete/Activate

Active Users

Username	Email	Role	Created At	Update	Ban	Delete
Admin	pp5104133@gmail.com	admin	2023-04-18 17:36:46	<button>Change</button>	<button>Ban</button>	<button>Delete</button>
BookishBibliophile	pp5104133@gmail.com	customer	2023-04-16 17:33:13	<button>Change</button>	<button>Ban</button>	<button>Delete</button>
BusinessBoss	pp5104133@gmail.com	customer	2023-04-16 17:38:35	<button>Change</button>	<button>Ban</button>	<button>Delete</button>
CosmicCrusader	pp5104133@gmail.com	customer	2023-04-16 17:44:23	<button>Change</button>	<button>Ban</button>	<button>Delete</button>
Test.123	pp5104133@gmail.com	customer	2023-05-12 12:33:48	<button>Change</button>	<button>Ban</button>	<button>Delete</button>
Vlada123	pp5104133@gmail.com	customer	2023-05-12 18:42:14	<button>Change</button>	<button>Ban</button>	<button>Delete</button>

Deleted Users

Username	Email	Role	Created At	Deleted At	Update	Activate
ArtisticAdventurer	pp5104133@gmail.com	customer	2023-04-16 17:28:54	2023-05-12 21:16:09	<button>Change</button>	<button>Activate</button>

Banned Users

Username	Email	Role	Created At	Banned At	Update	Unban
CultureCrusader	pp5104133@gmail.com	customer	2023-04-16 17:41:01	2023-05-12 21:16:18	<button>Change</button>	<button>Unban</button>

Admin-categories

Na ovoj stranici se nalaze informacije o kategorijama i podkategorijama.

Svaka kat/podkat sadrži informacije o svojoj roditeljskoj kategoriji kao i broj knjiga koje se nalaze u toj kategoriji

Status knjiga se može menjati a u pozadini se vrši soft delete

Ukoliko se izbriše kategorija, doćiće do brisanja i kategorija koje je referenciraju

Takođe se nalazi forma za unos kategorija i forma za unos podkategorija

Active Categories						
Name	Parent Category	Book Count	Created At	Update	Delete	
Mystery	/	28	2023-04-21 21:22:08	<button>Change</button>	<button>Delete</button>	
Travel	/	15	2023-04-21 21:25:35	<button>Change</button>	<button>Delete</button>	
Self Improvement	/	19	2023-04-21 21:26:32	<button>Change</button>	<button>Delete</button>	
Cooking	/	14	2023-04-21 21:27:16	<button>Change</button>	<button>Delete</button>	
Technology And Science	/	10	2023-04-22 07:18:49	<button>Change</button>	<button>Delete</button>	

Social History	History	12	2023-04-22 11:58:46	Change	Delete
Cultural History	History	13	2023-04-22 11:58:59	Change	Delete
Management And Leadership	Business And Economics	10	2023-04-22 11:59:55	Change	Delete
Finance And Accounting	Business And Economics	9	2023-04-22 12:00:13	Change	Delete
Horror Fiction	Fiction	4	2023-04-22 12:00:53	Change	Delete
Fantasy Fiction	Fiction	12	2023-04-22 12:01:07	Change	Delete

[Add Subcategory](#) [Add Category](#)

Deleted Categories

Name	Parent Category	Book Count	Deleted At	Update	Activate
Mystery	/	28	2023-05-12 21:31:58	Change	Activate
Detective Fiction	Mystery	10	2023-05-12 21:31:58	Change	Activate
Thrillers	Mystery	21	2023-05-12 21:31:58	Change	Activate

Parent Category

Mystery

Name

[Cancel](#)
[Save](#)

Name

[Cancel](#)
[Save](#)

Admin-publishers

Nalaze se informacije vezane za izdavače kao i broj knjiga koji oni izdaju.

Prilikom brisanja, u pozadini se vrši soft delete.

Active Publishers				
Name	Book Count	Created At	Update	Delete
Pottermore Publishing	7	2023-04-23 10:26:02	<button>Change</button>	<button>Delete</button>
Scribner	10	2023-04-23 10:26:13	<button>Change</button>	<button>Delete</button>
William Morrow Paperbacks	20	2023-04-23 10:26:21	<button>Change</button>	<button>Delete</button>
Grapevine	5	2023-04-23 10:26:30	<button>Change</button>	<button>Delete</button>
Mariner Books	4	2023-04-23 10:26:39	<button>Change</button>	<button>Delete</button>

Random House	10	2023-04-23 10:32:58	<button>Change</button>	<button>Delete</button>
Martins Press	5	2023-04-23 10:33:25	<button>Change</button>	<button>Delete</button>
Little Brown And Company	5	2023-04-23 10:33:43	<button>Change</button>	<button>Delete</button>
Norton And Company	5	2023-04-23 10:34:01	<button>Change</button>	<button>Delete</button>

Deleted Publishers				
Name	Book Count	Deleted At	Update	Activate

Admin-delivery-options

Active Delivery Types

Name	Orders Count	Created At	Update	Delete
Standard	14	2023-04-23 10:23:50	<button>Change</button>	<button>Delete</button>
Express	1	2023-04-23 10:24:12	<button>Change</button>	<button>Delete</button>
Next Day	0	2023-04-23 10:24:23	<button>Change</button>	<button>Delete</button>
Same Day	0	2023-04-23 10:24:32	<button>Change</button>	<button>Delete</button>

[Add Delivery Type](#)

Deleted Delivery Types

Name	Orders Count	Deleted At	Update	Activate
Test	0	2023-04-23 10:24:35	<button>Change</button>	<button>Activate</button>

[Activate Windows](#)
[Go to Settings to activate Windows](#)

Admin-survey

Active Surveys

Name	Created At	Option 1	Option 2	Option 3	Option 4	Option 5	Delete
website experience	2023-05-06 18:18:08	Terrible 2	Poor 1	Average 4	Good 2	Excellent 3	Delete

Finished Surveys

Name	Finished At	Option 1	Option 2	Option 3	Option 4	Option 5	Activate

Admin-messages

User Messages

Username	Email	Created At	Subject	Text
Admin	pp5104133@gmail.com	2023-05-10 14:04:52	Test	Test Test Test Test Test
Admin	pp5104133@gmail.com	2023-05-10 14:31:36	Lorem	<p>Lore ipsum is simply dummy text of the printing and typesetting industry. Lore ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem</p>
ArtisticAdventurer	pp5104133@gmail.com	2023-05-10 14:37:50	Proba	<p>Lore ipsum is simply dummy text of the printing and typesetting industry. Lore ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has</p>
Admin	pp5104133@gmail.com	2023-05-12 11:52:17	Test1	Test1 Test1 Test1 Test1 Test1

Admin-orders

User Orders

Username	Email	Finished At	Price	Status
Admin	pp5104133@gmail.com	2023-05-10 15:02:50	\$ 114.00	Pending
BusinessBoss	pp5104133@gmail.com	2023-05-12 14:30:26	\$ 15.00	Pending
ArtisticAdventurer	pp5104133@gmail.com	2023-05-12 13:53:52	\$ 15.00	Pending
Admin	pp5104133@gmail.com	2023-05-11 20:46:17	\$ 57.00	Pending
Admin	pp5104133@gmail.com	2023-05-12 11:53:29	\$ 197.00	Pending
Admin	pp5104133@gmail.com	2023-05-12 11:55:20	\$ 25.00	Pending

JoanneRowling	pp5104133@gmail.com	2023-05-12 13:45:18	\$ 14.00	Pending
BookishBibliophile	pp5104133@gmail.com	2023-05-12 13:54:55	\$ 29.00	Pending
Vlada123	pp5104133@gmail.com	2023-05-12 20:18:46	\$ 15.00	Pending
BusinessBoss	pp5104133@gmail.com	2023-05-10 08:18:40	\$ 43.00	Processing
CosmicCrusader	pp5104133@gmail.com	2023-05-10 08:19:51	\$ 43.00	Shipped
Admin	pp5104133@gmail.com	2023-05-09 23:15:26	\$ 84.00	Delivered

Admin-visitors

Visits

Page	Total visits	Total visits in %	Recent visits	Recent visits in %
home	13	32.5%	13	32.5%
shop	8	20%	8	20%
checkout	0	0%	0	0%
writer	0	0%	0	0%
book	5	12.5%	5	12.5%
signup	0	0%	0	0%
login	1	2.5%	1	2.5%
logout	1	2.5%	1	2.5%
user-profile	3	7.5%	3	7.5%

Admin-logged-in-users

Logged-in Users

Username	Email	Role	Date
Admin	pp5104133@gmail.com	admin	03-06-2023 18:56:17

Writer-books

Na stranici user-profile.php se nalazi link ka ovoj stranici kojeg može videti samo pisac ili admin koji je pristupio nalogu nekog pisca.

Nalaze se knjige koje napisao pisac i one se mogu obrisati ili aktivirati.

Active Books						
Image	Title	Price	Release Date	Update	Delete	
	Harry Potter and the Prisoner of Azkaban	\$15.00	2015-03-20 23:00:00	Change	Delete	
	Harry Potter and the Goblet of Fire	\$16.00	2015-12-07 23:00:00	Change	Delete	
	Harry Potter and the Half-Blood Prince	\$15.00	2015-02-28 23:00:00	Change	Delete	

Deleted Books						
Image	Title	Price	Deleted At	Update	Activate	
	The Ink Black Heart	\$23.00	2022-01-03 00:00:00	Change	Activate	

[Add Book](#)

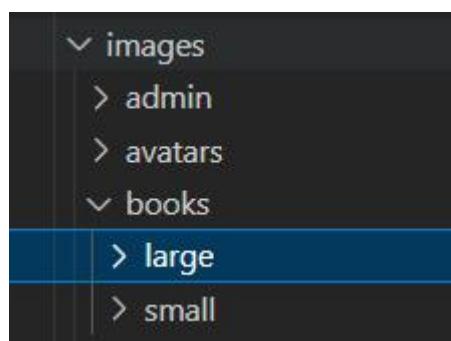
[Activate Windows](#)

Insert-book

Nalazi se forma za unos knjige.

Knjiga se može naći u više podkategorija

Nakon inserta slika kreiraju se dve slike različitih dimenzija



The image shows a 'Book Info' form. At the top, there is a placeholder image area with the text 'upload image of the book' and a 'Choose File' button. Below this, there are input fields for 'Book Title' (a text input), 'Page Count' (a text input), and 'Price' (a text input). In the bottom right corner of the form, there are links: 'Activate Window', 'Go to Settings to activ.', and 'Powered by '. The entire form is contained within a light gray box.

Book Title	
<input type="text"/>	
Page Count	Price
<input type="text"/>	<input type="text"/>
Release Date	Publisher
<input type="text"/> mm/dd/yyyy	<input type="button" value=""/>
Description	
<input type="text"/>	
Categories	
<ul style="list-style-type: none"> ▼ Mystery ▼ Travel ▼ Self Improvement ▼ Cooking <ul style="list-style-type: none"> <input type="checkbox"/> Healthy Cooking <input type="checkbox"/> Ethnic Cuisines ▼ Technology And Science 	
<ul style="list-style-type: none"> ▼ Travel <ul style="list-style-type: none"> <input checked="" type="checkbox"/> Adventure Travel <input type="checkbox"/> Cultural Travel ▼ Self Improvement ▼ Cooking ▼ Technology And Science ▼ Children's Books ▼ Romance <ul style="list-style-type: none"> <input checked="" type="checkbox"/> Historical Romance <input checked="" type="checkbox"/> Fantasy Romance ▼ History ▼ Business And Economics ▼ Fiction <ul style="list-style-type: none"> <input type="checkbox"/> Horror Fiction <input checked="" type="checkbox"/> Fantasy Fiction 	
<input type="button" value="Add Book"/>	

Update-book

Nalazi se forma za update informacija o knjizi koja je unapred popunjena sa odgovarajućim vrednostima.

Book Info



upload image of the book
 No file chosen

Book Title

Page Count

Price

Active
Go to Se

Book Title

Page Count Price

Release Date Publisher

Description

Categories

Mystery
 Detective Fiction
 Thrillers

Travel

Self Improvement

Cooking

Detective Fiction
 Thrillers

Travel

Self Improvement

Cooking

Technology And Science

Children's Books

Young Adult Fiction
 Humorous Stories

Romance

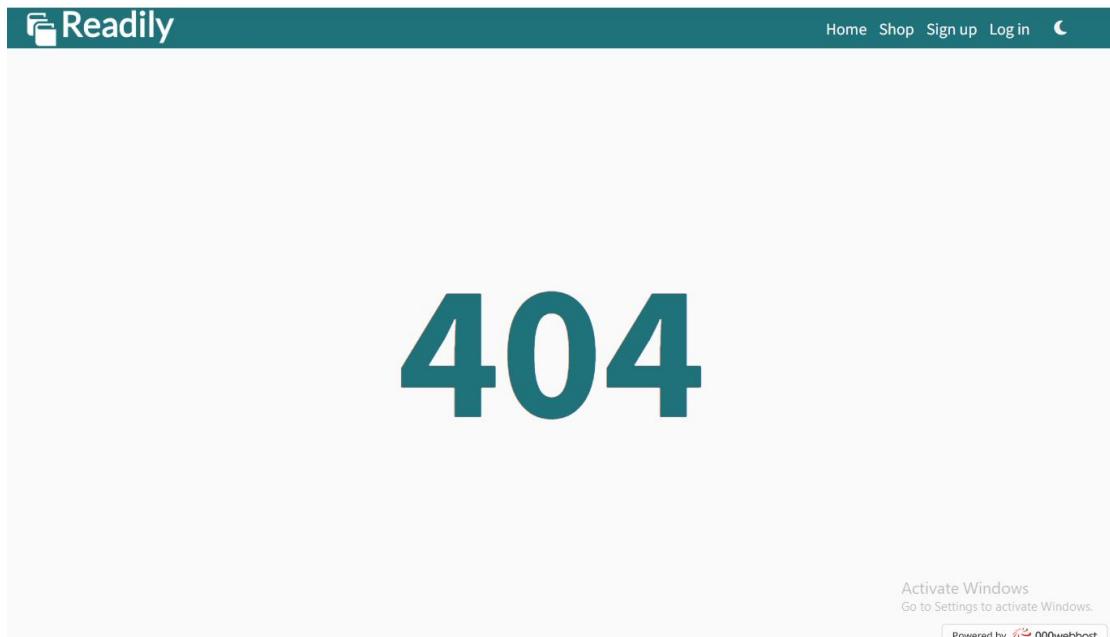
History

Business And Economics

Fiction

Horror Fiction
 Fantasy Fiction

404



Kodovi

.env

```
DATABASE=php_praktikum
SERVER=localhost
USERNAME=root
PASSWORD=
```

Config.php

```
<?php
define("BASE_URL", "http://127.0.0.1/Readilyphp/");
define("ABSOLUTE_PATH", $_SERVER["DOCUMENT_ROOT"] . "/Readilyphp");

define("ENV_FILE", ABSOLUTE_PATH . "/config/.env");
define("LOG_FILE", ABSOLUTE_PATH . "/data/log.txt");
define("LOGGED_ID_RECORD", ABSOLUTE_PATH . "/data/logged-in-
record.txt");
define("SEPARATOR", "\t");

define("SERVER", env("SERVER"));
define("DATABASE", env("DATABASE"));
define("USERNAME", env("USERNAME"));
define("PASSWORD", env("PASSWORD"));

function env($search){
    $data = file(ENV_FILE);
    $value = "";
    foreach($data as $data_row){
        list($name, $value) = explode("=", $data_row);

        if($name == $search) return trim($value);
    }
}

?>
```

Connection.php

```
<?php
require_once "config.php";

try{
    $conn = new PDO("mysql:host=" . SERVER . ";dbname=" . DATABASE .
";charset=utf8", USERNAME, PASSWORD);

    $conn->setAttribute(PDO::ATTR_DEFAULT_FETCH_MODE, PDO::FETCH_ASSOC);
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
}catch(PDOException $ex){
    echo $ex->getMessage();
}

function executeQuery($query){
    global $conn;
    return $conn->query($query)->fetchAll();
}

function executeQueryOneRow($query){
    global $conn;
    return $conn->query($query)->fetch();
}

function logAccess(){
    $open = fopen(LOG_FILE, "a");
    if($open){
        $date = date('d-m-Y H:i:s');
        fwrite($open,
"{$_SERVER['PHP_SELF']}\t{$date}\t{$_SERVER['REMOTE_ADDR']}\t\n");
        fclose($open);
    }
}
?>
```

Activate-book.php

```
<?php
    session_start();
    ob_start();
    header('Content-type: application/json');

    require_once "../../config/connection.php";
    require_once "../global/functions.php";
    require_once "../writer/functions.php";
    require_once "functions.php";

    $errorMessageArray = [];

    if($_SERVER['REQUEST_METHOD']=='GET'){
        http_response_code(404);
        redirect("../index.php?page=404");
        exit;
    }

    if(!isset($_SESSION['userId'])){
        http_response_code(401);
        $errorMessageArray[] = 'Access denied!';
        echo json_encode($errorMessageArray);
        exit;
    }

    if($_SESSION['userRoleName'] != 'admin' &&
$_SESSION['userRoleName'] != 'writer'){
        http_response_code(403);
        $errorMessageArray[] = 'Access denied!';
        echo json_encode($errorMessageArray);
        exit;
    }

    $bookId = $_POST['book-id'];
    $writerId = $_POST['writer-id'];

    try{
        activateBook($bookId);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
        exit;
    }
```

```
}

try{
    $allBooks = selectAllWriterBooksForTable($writerId);
}catch(PDOException $ex){
    http_response_code(500);
    $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
    echo json_encode($errorMessageArray);
    exit;
}

$message[] = $allBooks;
http_response_code(200);
echo json_encode($message);
ob_end_flush();

?>
```

Add-book.php

```
<?php
    session_start();
    ob_start();
    header('Content-type: application/json');

    require_once "../config/connection.php";
    require_once "../global/functions.php";
    require_once "../category/functions.php";
    require_once "../publisher/functions.php";
    require_once "../book/functions.php";

    if($_SERVER['REQUEST_METHOD']=='GET'){
        http_response_code(404);
        redirect("../index.php?page=404");
        exit;
    }

    $errorMessageArray = [];

    if(!isset($_SESSION['userId'])){
        http_response_code(401);
        $errorMessageArray[] = 'Access denied!';
        echo json_encode($errorMessageArray);
        exit;
    }

    if($_SESSION['userRoleName'] != 'admin' &&
$_SESSION['userRoleName'] != 'writer'){
        http_response_code(403);
        $errorMessageArray[] = 'Access denied!';
        echo json_encode($errorMessageArray);
        exit;
    }

    $writerId = $_POST['writer-id'];

    $bookTitle = $_POST['book-title-input'];
    $bookPrice = $_POST['price-input'];
    $pageCount = $_POST['page-count-input'];
    $publisherId = $_POST['publisher-input'];
    $releaseDate = $_POST['release-date-input'];
    $bookDescription = $_POST['book-description-input'];
    $selectedCategories = $_POST['book-category-cb'];
```

```

$bookImgName = $_FILES['book-image']['name'];
$bookImgTmpName = $_FILES['book-image']['tmp_name'];
$bookImgSize = $_FILES['book-image']['size'];
$bookImgType = pathinfo($_FILES['book-image']['name'],
PATHINFO_EXTENSION);

checkBookImg($bookImgTmpName,$bookImgSize,$bookImgType,$errorMessageArray);

checkDateInput($releaseDate,$errorMessageArray);

if(empty($bookTitle) || empty($bookPrice) || empty($pageCount) ||
empty($bookDescription) || !count($selectedCategories)){
    $errorMessageArray[] = "Must provide all values";
}

if($errorMessageArray){
    http_response_code(400);
    echo json_encode($errorMessageArray);
    exit;
}

try{
    $bookImgId = insertNewImageRow();
} catch(PDOException $ex){
    http_response_code(500);
    $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
    echo json_encode($errorMessageArray);
    exit;
}

$smallImageHeight = 200;
$result = createNewBookImage($bookImgTmpName, $smallImageHeight,
$bookImgId, $bookImgType, "small/");

if(!$result){
    http_response_code(500);
    $errorMessageArray[] = 'Server error.';
    echo json_encode($errorMessageArray);
    exit;
}

$largeImageHeigth = 400;
$result = createNewBookImage($bookImgTmpName, $largeImageHeigth,
$bookImgId, $bookImgType, "large/");

if(!$result){
    http_response_code(500);
}

```

```

$errorMessageArray[] = 'Server error.';
echo json_encode($errorMessageArray);
exit;
}

//imagedestroy($originalImage);

try{
    $bookId = insertNewBook($bookTitle, $pageCount, $bookPrice,
$releaseDate, $bookDescription, $publisherId, $writerId, $bookImgId);
}catch(PDOException $ex){
    http_response_code(500);
    $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
    echo json_encode($errorMessageArray);
    exit;
}

try{
    updateBookImageRowName($bookImgId,$bookImgType);
}catch(PDOException $ex){
    http_response_code(500);
    $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
    echo json_encode($errorMessageArray);
    exit;
}

try{
    insertBookCategories($bookId,$selectedCategories);
}catch(PDOException $ex){
    http_response_code(500);
    $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
    echo json_encode($errorMessageArray);
    exit;
}

$message[] = 'Book was successfully added!';
http_response_code(200);
echo json_encode($message);
ob_end_flush();
?>
```

Book-rating.php

```
<?php
    session_start();
    ob_start();
    header('Content-type: application/json');

    require_once "../config/connection.php";
    require_once "../global/functions.php";
    require_once "functions.php";
    require_once "../user/functions.php";

    if($_SERVER['REQUEST_METHOD']=='GET'){
        http_response_code(404);
        redirect("../index.php?page=404");
        exit;
    }

    if(!isset($_SESSION['userId'])){
        http_response_code(401);
        $errorMessageArray[] = 'Access denied!';
        echo json_encode($errorMessageArray);
        exit;
    }

    $userId = $_SESSION['userId'];
    $bookId = $_POST['book-id'];
    $bookStars = $_POST['book-stars'];

    try{
        $reviewId = getSingleReview($userId,$bookId);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
        exit;
    }

    if(!$reviewId){
        try{
            $reviewId = insertReview($userId,$bookId,$bookStars+1);
        }catch(PDOException $ex){
            http_response_code(500);
            $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
            echo json_encode($errorMessageArray);
        }
    }
}
```

```
        exit;
    }

    $inform = "Your rating has been successfully saved.";
    $messsage[] = $inform;
    http_response_code(200);
    echo json_encode($messsage);
    exit;
}

try{
    updateReview($reviewId['id'],$bookStars+1);
}catch(PDOException $ex){
    http_response_code(500);
    $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
    echo json_encode($errorMessageArray);
    exit;
}

$inform = "Your rating has been successfully saved.";
$messsage[] = $inform;
http_response_code(200);
echo json_encode($messsage);
ob_end_flush();
?>
```

Delete-book.php

```
<?php
    session_start();
    ob_start();
    header('Content-type: application/json');

    require_once "../../config/connection.php";
    require_once "../global/functions.php";
    require_once "../writer/functions.php";
    require_once "functions.php";

    if($_SERVER['REQUEST_METHOD']=='GET'){
        http_response_code(404);
        redirect("../index.php?page=404");
        exit;
    }

    $errorMessageArray = [];

    if(!isset($_SESSION['userId'])){
        http_response_code(401);
        $errorMessageArray[] = 'Access denied!';
        echo json_encode($errorMessageArray);
        exit;
    }

    if($_SESSION['userRoleName'] != 'admin' &&
$_SESSION['userRoleName'] != 'writer'){
        http_response_code(403);
        $errorMessageArray[] = 'Access denied!';
        echo json_encode($errorMessageArray);
        exit;
    }

    $bookId = $_POST['book-id'];

    try{
        deleteBook($bookId);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
        exit;
    }
```

```
try{
    $writerId = $_POST['writer-id'];
    $allBooks = selectAllWriterBooksForTable($writerId);
}catch(PDOException $ex){
    http_response_code(500);
    $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
    echo json_encode($errorMessageArray);
    exit;
}

$message[] = $allBooks;
http_response_code(200);
echo json_encode($message);
ob_end_flush();

?>
```

Models > book > functions.php

```
<?php

function getBestSellingBooks(){
    global $conn;

    $query = "SELECT book.name AS title, COUNT(order_book.book_id) AS
book_count,
    image.src AS src, book.id AS id, CONCAT(user.first_name,'
',user.last_name) AS writer,
    book.user_id AS writer_id, IFNULL(CEIL(AVG(review.stars)),0) AS
review
    FROM book
    LEFT JOIN order_book ON order_book.book_id = book.id
    JOIN image ON image.id = book.image_id
    JOIN user ON user.id = book.user_id
    LEFT JOIN review ON review.book_id = book.id
    JOIN publisher ON publisher.id = book.publisher_id
    WHERE book.status = 1 AND publisher.status = 1
    GROUP BY book.id
    ORDER BY book_count DESC
    LIMIT 20";

    $statement = $conn->prepare($query);
    $statement->execute();

    $result = $statement->fetchAll();

    return $result;
}

function selectRelatedBooksToACategories($categories){
    global $conn;

    $parentIdValues = "".$categories[0]['id'];

    for($i=1;$i<count($categories);$i++){
        $parentIdValues .= ", " . $categories[$i]['id'];
    }

    $query =
    "SELECT book.name AS title, image.src AS src, book.id AS id,
CONCAT(user.first_name,' ',user.last_name) AS writer,
```

```

        book.user_id AS writer_id, IFNULL(CEIL(AVG(review.stars)),0) AS
review
    FROM book
    JOIN image ON image.id = book.image_id
    JOIN user ON user.id = book.user_id
    LEFT JOIN review ON review.book_id = book.id
    JOIN publisher ON publisher.id = book.publisher_id
    JOIN book_category ON book_category.book_id = book.id
    JOIN category child ON child.id = book_category.category_id
    JOIN category parent ON parent.id = child.category_id
    WHERE parent.id IN ($parentIdValues) AND book.status = 1 AND
publisher.status = 1
    GROUP BY book.id
    ORDER BY RAND()
    LIMIT 20";

$stmt = $conn->prepare($query);
$stmt->execute();

$result = $stmt->fetchAll();

return $result;
}

function displayBookArticle($book){
    $lastDigit = substr($book['id'], -1);
    $display = "<article class='article-book'>
                    <div class='bg-article-color-".$lastDigit."
article-div-img-container'>
                        <img class='set-brightness'
src='assets/images/books/small/" . $book['src'] . "'"
alt='".$book['title']."'>
                    </div>
                    <div class='article-books-text-container'>
                        <div class='title-and-author'>
                            <h3>".$book['title']. "</h3>
                            <a class='author-link'
href='index.php?page=writer&id=" . $book['writer_id'] ."'>" .
$book['writer'] . "</a>
                        </div>
                        <div class='stars-and-cart-container'>
                            <i class='fa-solid fa-cart-shopping shopping-
cart' id='book-id-' . $book['id'] . "'></i>
                            <div class='stars-container'>
";
    for($i=0;$i<5;$i++){
        if($book['review']>$i){

```

```

        $display .= "<i class='fa-solid fa-
star'></i>";
            continue;
        }else{
            $display .= "<i class='fa-regular fa-
star'></i>";
        }

    }

$display .=

    "
</div>
<p class='rating-text'>";

if($book['review']){
    $display .= $book['review'] . "/5";
}else{
    $display .= "0 ratings";
}

$display .=
    "</p>
    </div>
    </div>
    <a class='link-to-single-a-book'
href='index.php?page=book&id=".$book['id']."'>
        </a>
    </article>";

return $display;
}

function getBooksForShop($filter,$sort){
    global $conn;

    $query =
        "SELECT DISTINCT book.name AS 'title' , COUNT(order_book.book_id)
AS 'book_count',book.price AS 'price',book.release_date AS release_date,
        image.src AS 'src', book.id AS 'id', CONCAT(user.first_name,
',user.last_name) AS 'writer',
        book.user_id AS 'writer_id', IFNULL(CEIL(AVG(review.stars)),0)  AS
'review'
        FROM book
        LEFT JOIN order_book ON order_book.book_id = book.id
        JOIN image ON image.id = book.image_id
        JOIN user ON user.id = book.user_id

```

```

        JOIN publisher ON publisher.id = book.publisher_id
        JOIN book_category ON book_category.book_id = book.id
        LEFT JOIN review ON review.book_id = book.id
        WHERE book.status = 1 AND user.status = 1 AND publisher.status = 1
$filter
    GROUP BY book.id , book_category.category_id
    ORDER BY $sort";

$stmt = $conn->prepare($query);
$stmt->execute();

$result = $stmt->fetchAll();

return $result;
}

function selectBook($id){
    global $conn;

$query =
"SELECT book.id AS id, book.name as title,
CONCAT(user.first_name, ' ', user.last_name) AS writer,
image.src AS src, book.description as description, book.price as
price,
book.release_date as release_date, publisher.name as publisher_name,
book.page_count as page_count,
IFNULL(CEIL(AVG(review.stars)),0) AS review,
(
    SELECT COUNT(review.book_id)
    FROM review
    WHERE review.book_id = :book_id
) AS review_count,
(
    SELECT COUNT(comment.user_id)
    FROM comment
    WHERE comment.book_id = :book_id
) AS comment_count,
user.id AS writer_id
FROM book
LEFT JOIN review ON review.book_id = book.id
LEFT JOIN comment ON comment.book_id = book.id
JOIN user ON book.user_id = user.id
JOIN publisher ON publisher.id = book.publisher_id
JOIN image ON image.id = book.image_id
WHERE book.id = :book_id AND book.status = 1 AND publisher.status =
1 AND user.status = 1
GROUP BY book.id";

$stmt = $conn->prepare($query);

```

```

        $statement->bindParam(':book_id',$id);
        $statement->execute();

        $result = $statement->fetch();
        return $result;
    }

function getBookComments($bookId){
    global $conn;

    $query =
        'SELECT comment.id, comment.comment_text , user.username,
IFNULL(avatar.src,"default-avatar.jpg") as avatar,
IFNULL(review.stars,0) as stars
        FROM comment
        JOIN user ON user.id = comment.user_id
        JOIN book ON book.id = comment.book_id
        LEFT JOIN avatar ON avatar.user_id = user.id
        LEFT JOIN review ON user.id = review.user_id AND review.book_id =
book.id
        WHERE book.id = :book_id';

    $statement = $conn->prepare($query);
    $statement->bindParam(':book_id',$bookId);
    $statement->execute();

    $result = $statement->fetchAll();
    return $result;
}

function getSingleReview($userId,$bookId){
    global $conn;

    $query =
        "SELECT id , stars
        FROM review
        WHERE review.user_id = :user_id AND review.book_id = :book_id";
    $statement = $conn->prepare($query);
    $statement->bindParam(':user_id',$userId);
    $statement->bindParam(':book_id',$bookId);

    $statement->execute();
    $result = $statement->fetch();
    return $result;
}

function getImagesForComment($commentId){
    global $conn;

```

```

$query =
"SELECT * FROM image
JOIN comment_image ON comment_image.image_id = image.id
WHERE comment_image.comment_id = :comment_id";

$statement = $conn->prepare($query);
$statement->bindParam(":comment_id",$commentId);
$statement->execute();

$result = $statement->fetchAll();
return $result;
}

function selectBookInformation($bookId){
    global $conn;

    $query =
    'SELECT book.id AS id,book.user_id AS writer_id, book.name AS name,
book.page_count AS page_count, book.price AS price, book.publisher_id
AS publisher_id, book.release_date AS release_date, book.description AS
description, image.src AS src FROM book
    JOIN image ON book.image_id = image.id
    WHERE book.id = :book_id';

    $statement = $conn->prepare($query);
    $statement->bindParam(':book_id',$bookId);

    $statement->execute();

    $result = $statement->fetch(PDO::FETCH_OBJ);

    return $result;
}

function selectBookActiveCategories($bookId){
    global $conn;

    $query =
    'SELECT book_category.category_id FROM book_category JOIN category
ON category.id = book_category.category_id
    WHERE category.status = 1 AND book_category.book_id = :book_id';

    $statement = $conn->prepare($query);
    $statement->bindParam(':book_id',$bookId);

    $statement->execute();

    $result = $statement->fetchAll();
}

```

```

        return $result;
    }

function checkBookImg($name,$size,$type,&$errorMessagesArray){
    $allowedTypes = array('jpg', 'jpeg', 'png');

    if(!in_array($type, $allowedTypes)) {
        $errorMessagesArray[] = 'File type not allowed. Only JPG, JPEG, and PNG files are allowed.';
        return;
    }

    if($size > 700 * 1024) {
        $errorMessagesArray[] = 'File size exceeds 700KB limit.';
        return;
    }

    list($width, $height) = getimagesize($name);

    if($height/$width < 200/170){
        $errorMessagesArray[] = 'Width must be lesser than height.';
        return;
    }
}

function checkDateInput($releaseDate,&$errorMessageArray){
    $dateTimestamp = strtotime($releaseDate);
    $currentTimestamp = time();
    if ($dateTimestamp > $currentTimestamp) {
        $errorMessageArray = "The date is in the future!";
    }
}

function getBookImage($bookId){
    global $conn;

    $query = 'SELECT image.src AS src, image.id AS id FROM image JOIN book ON image.id = book.image_id WHERE book.id = :book_id';
    $statement = $conn->prepare($query);

    $statement->bindParam(':book_id',$bookId);

    $statement->execute();

    $result = $statement->fetch(PDO::FETCH_OBJ);

    return $result;
}

```

```

function updateBookImageRowName($bookImgId,$avatarType){
    global $conn;

    $src = $bookImgId . '.' . $avatarType;

    $query = 'UPDATE image SET src = :src WHERE id = :book_img_id';
    $statement = $conn->prepare($query);

    $statement->bindParam(":src",$src);
    $statement->bindParam(":book_img_id",$bookImgId);
    $statement->execute();

}

function updateBook($bookTitle, $pageCount, $bookPrice, $releaseDate,
$bookDescription, $publisherId, $bookId){
    global $conn;

    $query =
'UPDATE book
SET name = :book_title, page_count = :page_count,
price = :book_price, release_date = :release_date,
description = :book_description, publisher_id = :publisher_id
WHERE id = :book_id';

    $statement = $conn->prepare($query);

    $statement->bindParam(':book_title',$bookTitle);
    $statement->bindParam(':page_count',$pageCount);
    $statement->bindParam(':book_price',$bookPrice);
    $statement->bindParam(':release_date',$releaseDate);
    $statement->bindParam(':book_description',$bookDescription);
    $statement->bindParam(':publisher_id',$publisherId);
    $statement->bindParam(':book_id',$bookId);

    $statement->execute();
}

function deleteBook($bookId){
    global $conn;

    $query = 'UPDATE book SET status = 0 WHERE id = :book_id';
    $statement = $conn->prepare($query);

    $statement->bindParam(':book_id',$bookId);

    $statement->execute();
}

```

```

function activateBook($bookId){
    global $conn;

    $query = 'UPDATE book SET status = 1 WHERE id = :book_id';
    $statement = $conn->prepare($query);

    $statement->bindParam(':book_id', $bookId);

    $statement->execute();
}

function insertNewImageRow(){
    global $conn;

    $query = "INSERT INTO image () VALUES ()";
    $statement = $conn->prepare($query);

    $statement->execute();

    $lastInsertId = $conn->lastInsertId();

    return $lastInsertId;
}

function insertNewBook($bookTitle, $pageCount, $bookPrice, $releaseDate,
$bookDescription, $publisherId, $writerId, $bookImgId){
    global $conn;

    $query =
        'INSERT INTO book (name, page_count, price, release_date,
description, publisher_id, user_id, image_id)
        VALUES(:book_title, :page_count, :book_price, :release_date, :book_
description, :publisher_id, :writer_id, :book_img_id )';
    $statement = $conn->prepare($query);

    $statement->bindParam(':book_title', $bookTitle);
    $statement->bindParam(':page_count', $pageCount);
    $statement->bindParam(':book_price', $bookPrice);
    $statement->bindParam(':release_date', $releaseDate);
    $statement->bindParam(':publisher_id', $publisherId);
    $statement->bindParam(':writer_id', $writerId);
    $statement->bindParam(':book_img_id', $bookImgId);
    $statement->bindParam(':book_description', $bookDescription);

    $statement->execute();

    $lastInsertId = $conn->lastInsertId();
}

```

```

        return $lastInsertId;
    }

function insertBookCategories($bookId,$selected_categories){
    global $conn;

    foreach($selected_categories as $category){
        $query = 'INSERT INTO book_category (book_id,category_id)
VALUES (:book_id,:category)';
        $statement = $conn->prepare($query);

        $statement->bindParam(':book_id',$bookId);
        $statement->bindParam(':category',$category);

        $statement->execute();
    }
}

function insertReview($userId,$bookId,$bookStars){
    global $conn;

    $query =
        "INSERT INTO review (stars, user_id, book_id)
VALUES(:book_stars, :user_id, :book_id)";
    $statement = $conn->prepare($query);
    $statement->bindParam(":book_stars",$bookStars);
    $statement->bindParam(":user_id",$userId);
    $statement->bindParam(":book_id",$bookId);

    $statement->execute();

    $lastInsertId = $conn->lastInsertId();

    return $lastInsertId;
}

function updateReview($reviewId,$bookStars){
    global $conn;

    $query = "UPDATE review SET stars = :stars WHERE id = :id";
    $statement = $conn->prepare($query);
    $statement->bindParam(":stars",$bookStars);
    $statement->bindParam(":id",$reviewId);

    $statement->execute();
}

```

```

function insertCommentText($userId,$bookId,$comment){
    global $conn;

    $query = "INSERT INTO comment(comment_text,user_id,book_id)
VALUES(:comment,:user_id,:book_id)";
    $statement = $conn->prepare($query);
    $statement->bindParam(':comment',$comment);
    $statement->bindParam(':user_id',$userId);
    $statement->bindParam(':book_id',$bookId);

    $statement->execute();

    $lastInsertId = $conn->lastInsertId();

    return $lastInsertId;
}

function insertCommentImageRow($commentImgId,$commentId){
    global $conn;

    $query = "INSERT INTO comment_image(comment_id,image_id)
VALUES(:comment_id,:comment_img_id)";
    $statement = $conn->prepare($query);
    $statement->bindParam(':comment_id',$commentId);
    $statement->bindParam(':comment_img_id',$commentImgId);
    $statement->execute();
}

function createNewBookImage($bookImgTmpName, $height, $bookImgId,
$bookImgType, $folder){
    $uploadDir = '../../assets/images/books/' . $folder;
    $filePath = $uploadDir . $bookImgId . '.' . $bookImgType;

    if ($bookImgType === 'jpg' || $bookImgType === 'jpeg') {
        $originalImage = imagecreatefromjpeg($bookImgTmpName);
    } elseif ($bookImgType === 'png') {
        $originalImage = imagecreatefrompng($bookImgTmpName);
    }

    $originalWidth = imagesx($originalImage);
    $originalHeight = imagesy($originalImage);

    $newWidth = round(($originalWidth / $originalHeight) * $height);

    $newImage = imagecreatetruecolor($newWidth, $height);

    imagecopyresampled($newImage, $originalImage, 0, 0, 0, 0, $newWidth,
$height, $originalWidth, $originalHeight);
}

```

```
$quality = $folder == "small/" ? 75 : 100;

if ($bookImgType === 'jpg' || $bookImgType === 'jpeg') {
    $result = imagejpeg($newImage, $filePath, $quality);
} elseif ($bookImgType === 'png') {
    $result = imagepng($newImage, $filePath, $quality);
}

imagedestroy($newImage);
imagedestroy($originalImage);

return $result;
}
?>
```

Get-book-active-categories.php

```
<?php
    session_start();
    ob_start();
    header('Content-type: application/json');

    require_once "../../config/connection.php";
    require_once "../global/functions.php";
    require_once "functions.php";

    $errorMessages = [];

    if(!isset($_SESSION["userRoleName"]) || 
    ($_SESSION["userRoleName"] != "admin") && $_SESSION["userRoleName"] != 
    "writer") || !isset($_GET['book-id'])){
        http_response_code(403);
        $errorMessages[] = 'Access denied!';
        echo json_encode($errorMessages);
        exit;
    }

    $bookId = $_GET['book-id'];

    $currentBookCategories = selectBookActiveCategories($bookId);

    http_response_code(200);
    echo json_encode($currentBookCategories);
    ob_end_flush();
?>
```

Get-suggested-books.php

```
<?php
    session_start();
    ob_start();
    header('Content-type: application/json');

    require_once "../config/connection.php";
    require_once "../global/functions.php";
    require_once "../user/functions.php";
    require_once "../category/functions.php";
    require_once "functions.php";

    if($_SERVER['REQUEST_METHOD']=='GET'){
        http_response_code(404);
        redirect("../index.php?page=404");
        exit;
    }

    $errorMessageArray = [];

    $preferredCategories = $_POST['checkbox-preferred-categories'];
    $userId = $_SESSION['userId'];

    $associativeArray = array_map(function($value) {
        return array('id' => $value);
    }, $preferredCategories);

    try{
        $suggestedBooks =
selectRelatedBooksToACategories($associativeArray);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
        exit;
    }

    shuffle($suggestedBooks);

    try{
        insertUsersPreferredCategories($preferredCategories,$userId);
    }catch(PDOException $ex){
```

```
http_response_code(500);
$errorMessageArray[] = 'Server error: ' . $ex->getMessage();
echo json_encode($errorMessageArray);
exit;
}

setcookie('hasPreferredCategories', true, time() + 36000, '/');

http_response_code(200);
echo json_encode($suggestedBooks);
ob_end_flush();
?>
```

Insert-comment.php

```
<?php
    session_start();
    ob_start();
    header('Content-type: application/json');

    require_once "../../config/connection.php";
    require_once "../../global/functions.php";
    require_once "functions.php";
    require_once "../../user/functions.php";

    if($_SERVER['REQUEST_METHOD']=='GET'){
        http_response_code(404);
        redirect("../index.php?page=404");
        exit;
    }

    $errorMessageArray = [];

    if(!isset($_SESSION['userId'])){
        http_response_code(401);
        $errorMessageArray[] = 'Access denied!';
        echo json_encode($errorMessageArray);
        exit;
    }

    $userId = $_SESSION['userId'];
    $bookId = $_POST['book-id'];
    $comment = $_POST['comment-input'];

    checkInputElementWithRegex('/(\s.*){4,}/',$comment,'Comment must
have at least 5 words.',$errorMessageArray);

    $totalSize = 0;
    $maxSize = 2 * 1024 * 1024;
    for($i=0;$i<count($_FILES['comment-image']['name']);$i++){

        if(empty($_FILES['comment-image']['name'][$i])){
            continue;
        }

        $imageName = $_FILES['comment-image']['name'][$i];
```

```

$imageTmpName = $_FILES['comment-image']['tmp_name'][$i];
$imageSize = $_FILES['comment-image']['size'][$i];
$imageType = pathinfo($_FILES['comment-image']['name'][$i],
PATHINFO_EXTENSION);

$allowedTypes = array('jpg', 'jpeg', 'png');

if(!in_array($imageType, $allowedTypes)) {
    $errorMessageArray[] = 'File type not allowed. Only JPG,
JPEG, and PNG files are allowed.';
}
$totalSize += $imageSize;
}

if($totalSize > $maxSize){
    $errorMessageArray[] = 'The total size of the uploaded files
exceeds the maximum allowed size of 2 MB.';
}

if($errorMessageArray){
    http_response_code(400);
    echo json_encode($errorMessageArray);
    exit;
}

try{
    $comment_id = insertCommentText($userId,$bookId,$comment);
}catch(PDOException $ex){
    http_response_code(500);
    $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
    echo json_encode($errorMessageArray);
    exit;
}

for($i=0;$i<count($_FILES['comment-image']['name']);$i++){
    if(empty($_FILES['comment-image']['name'][$i])){
        continue;
    }

    $imageName = $_FILES['comment-image']['name'][$i];
    $imageTmpName = $_FILES['comment-image']['tmp_name'][$i];
    $imageSize = $_FILES['comment-image']['size'][$i];
    $imageType = pathinfo($_FILES['comment-image']['name'][$i],
PATHINFO_EXTENSION);

    try{
        $commentImgId = insertNewImageRow();
    }catch(PDOException $ex){

```

```

        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
        exit;
    }

    $uploadDir = ' ../../assets/images/comments/';
    $filePath = $uploadDir . $commentImgId . '.' . $imageType;
    $result = move_uploaded_file($imageTmpName,$filePath);

    if(!$result){
        http_response_code(500);
        $errorMessageArray[] = 'Server error.' ;
        echo json_encode($errorMessageArray);
        exit;
    }

    try{
        updateBookImageRowName($commentImgId,$imageType);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
        exit;
    }

    try{
        insertCommentImageRow($commentImgId,$comment_id);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
        exit;
    }
}

http_response_code(200);
$message[] = 'Your comment has been successfully uploaded.';
echo json_encode($message);
ob_end_flush();
?>
```

Update-book.php

```
<?php
    session_start();
    ob_start();
    header('Content-type: application/json');

    require_once "../../config/connection.php";
    require_once "../../global/functions.php";
    require_once "../../category/functions.php";
    require_once "functions.php";

    if($_SERVER['REQUEST_METHOD']=='GET'){
        http_response_code(404);
        redirect("../index.php?page=404");
        exit;
    }

    $errorMessageArray = [];

    if(!isset($_SESSION['userId'])){
        http_response_code(401);
        $errorMessageArray[] = 'Access denied!';
        echo json_encode($errorMessageArray);
        exit;
    }

    if($_SESSION['userRoleName'] != 'admin' &&
    $_SESSION['userRoleName'] != 'writer'){
        http_response_code(403);
        $errorMessageArray[] = 'Access denied!';
        echo json_encode($errorMessageArray);
        exit;
    }

    $bookId = $_POST['book-id'];

    $bookTitle = $_POST['book-title-input'];
    $bookPrice = $_POST['price-input'];
    $pageCount = $_POST['page-count-input'];
    $publisherId = $_POST['publisher-input'];
    $releaseDate = $_POST['release-date-input'];
    $bookDescription = $_POST['book-description-input'];
```

```

$selectedCategories = $_POST['book-category-cb'];

$bookImgName = $_FILES['book-image']['name'];
$bookImgTmpName = $_FILES['book-image']['tmp_name'];
$bookImgSize = $_FILES['book-image']['size'];
$bookImgType = pathinfo($_FILES['book-image']['name'],
PATHINFO_EXTENSION);

try{
    $selectedCategoriesDb = selectBookActiveCategories($bookId);
}catch(PDOException $ex){
    http_response_code(500);
    $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
    echo json_encode($errorMessageArray);
    exit;
}

$preveouslySelectedCategories = [];

for($i=0;$i<count($selectedCategoriesDb);$i++){
    $preveouslySelectedCategories[] =
$selectedCategoriesDb[$i]['category_id'];
}

$hasBookImg = false;
if(!empty($bookImgName)){
    checkBookImg($bookImgTmpName,$bookImgSize,$bookImgType,$errorMessageArray);
    $hasBookImg = true;
}

checkDateInput($releaseDate,$errorMessageArray);

if(empty($bookTitle) || empty($bookPrice) || empty($pageCount) ||
empty($bookDescription) || !count($selectedCategories)){
    $errorMessageArray[] = "Must proveide all values";
}

if($errorMessageArray){
    http_response_code(400);
    echo json_encode($errorMessageArray);
    exit;
}

if($hasBookImg){
    try{
        $bookImgObj = getBookImage($bookId);
    }catch(PDOException $ex){
}

```

```

        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
        exit;
    }

    $uploadDir = '.../.../assets/images/books/';
    $filePath = $uploadDir . $bookImgObj->src;
    unlink($filePath);

    $smallImageHeight = 200;
    $result = createNewBookImage($bookImgTmpName, $smallImageHeight,
$bookImgObj->id, $bookImgType, "small/");

    if(!$result){
        http_response_code(500);
        $errorMessageArray[] = 'Server error.';
        echo json_encode($errorMessageArray);
        exit;
    }

    $largeImageHeigth = 400;
    $result = createNewBookImage($bookImgTmpName, $largeImageHeigth,
$bookImgObj->id, $bookImgType, "large/");

    if(!$result){
        http_response_code(500);
        $errorMessageArray[] = 'Server error.';
        echo json_encode($errorMessageArray);
        exit;
    }

    // $filePath = $uploadDir . $bookImgObj->id . '.' . $bookImgType;
    // $result = move_uploaded_file($bookImgTmpName,$filePath);

    try{
        updateBookImageRowName($bookImgObj->id,$bookImgType);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
        exit;
    }
}

try{
    updateBook($bookTitle, $pageCount, $bookPrice, $releaseDate,
$bookDescription, $publisherId, $bookId);
}catch(PDOException $ex){

```

```

        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
        exit;
    }

    $addedCategories = array_diff($selectedCategories,
$preveouslySelectedCategories);
    $deletedCategories = array_diff($preveouslySelectedCategories,
$selectedCategories);

    $addedCategories = array_values($addedCategories);
    $deletedCategories = array_values($deletedCategories);

try{
    addUpdatedBookCategories($bookId,$addedCategories);
}catch(PDOException $ex){
    http_response_code(500);
    $errorMessageArray[] = $addedCategories;
    $errorMessageArray[] = $deletedCategories;
    $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
    echo json_encode($errorMessageArray);
    exit;
}

try{
    deleteUpdatedBookCategories($bookId,$deletedCategories);
}catch(PDOException $ex){
    http_response_code(500);
    $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
    echo json_encode($errorMessageArray);
    exit;
}

$message[] = $_POST['writer-id'];
http_response_code(200);
echo json_encode($messsage);
ob_end_flush();
?>
```

Cart-logic.php

```
<?php
    session_start();
    ob_start();
    header('Content-type: application/json');

    require_once "../config/connection.php";
    require_once "../book/functions.php";
    require_once "../global/functions.php";
    require_once "../user/functions.php";
    require_once "functions.php";

    if($_SERVER['REQUEST_METHOD']=='GET'){
        http_response_code(404);
        redirect("../index.php?page=404");
        exit;
    }

    if(!isset($_SESSION['userId'])){
        http_response_code(401);
        $errorMessageArray[] = 'Access denied!';
        echo json_encode($errorMessageArray);
        exit;
    }

    $userId = $_SESSION['userId'];

    $cookieValue = $_COOKIE['cart'];
    $cookieCart = json_decode($cookieValue);

    try{
        $databaseCart = getCartUserInfo($userId);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
        exit;
    }

    $cookieCartIds = array_column($cookieCart, "id");
    $databaseCartIds = array_column($databaseCart, "id");
```

```

$itemsToDelete = array_diff($databaseCartIds, $cookieCartIds);

try{
    deleteUserCartItems($userId,$itemsToDelete);
}catch(PDOException $ex){
    http_response_code(500);
    $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
    echo json_encode($errorMessageArray);
    exit;
}

$itemsToInsertIds = array_diff($cookieCartIds,$databaseCartIds);

$booksToInsert = array();
foreach ($cookieCart as $book) {
    if (in_array($book->id, $itemsToInsertIds)) {
        $booksToInsert[] = $book;
    }
}

try{
    insertUserCartItems($userId,$booksToInsert);
}catch(PDOException $ex){
    http_response_code(500);
    $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
    echo json_encode($errorMessageArray);
    exit;
}

$itemsToUpdateIds =
array_intersect($cookieCartIds,$databaseCartIds);

$booksToUpdate = array();
foreach ($cookieCart as $book) {
    if (in_array($book->id, $itemsToUpdateIds)) {
        $booksToUpdate[] = $book;
    }
}

try{
    updatetUserCartItems($userId,$booksToUpdate);
}catch(PDOException $ex){
    http_response_code(500);
    $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
    echo json_encode($errorMessageArray);
    exit;
}

```

```
$booksInformation = [];
foreach($cookieCart as $bookInCart){
    $bookInformation = selectBook($bookInCart->id);

    $bookInformationObject = (object) array(
        'id' => $bookInformation['id'],
        'title' => $bookInformation['title'],
        'src' => $bookInformation['src'],
        'price' => $bookInformation['price'],
        'quantity' => $bookInCart->quantity
    );

    $booksInformation[] = $bookInformationObject;
}

$message = $booksInformation;
http_response_code(200);
echo json_encode($message);
ob_end_flush();
?>
```

Models > cart > functions.php

```
<?php

function insertNewUserOrder($userId){
    global $conn;

    $query = "INSERT INTO user_order (user_id) VALUES (:user_id)";
    $statement = $conn->prepare($query);
    $statement->bindParam(':user_id',$userId);

    $statement->execute();
    return $conn->lastInsertId();
}

function selectUserCart($userId,$orderId){
    global $conn;

    $query =
"SELECT order_book.book_id as id, order_book.quantity as quantity
FROM order_book
JOIN user_order ON user_order.id = order_book.order_id
JOIN user ON user.id = user_order.user_id
WHERE user_order.user_id = :user_id AND user_order.id = :order_id";

    $statement = $conn->prepare($query);
    $statement->bindParam(':user_id',$userId);
    $statement->bindParam(':order_id',$orderId);
    $statement->execute();

    $result = $statement->fetchAll();

    return $result;
}

function getCartUserInfo($userId){
    global $conn;

    $query =
"SELECT order_book.book_id as id, order_book.quantity as quantity,
order_book.unit_price as price
FROM order_book
JOIN user_order ON user_order.id = order_book.order_id"
```

```

        WHERE user_order.finished_at IS NULL AND user_order.user_id
= :user_id";

        $statement = $conn->prepare($query);
        $statement->bindParam(':user_id',$userId);

        $statement->execute();
        $result = $statement->fetchAll();
        return $result;
    }

function updateCartItems($userId,$booksToUpdate){
    global $conn;

    $query =
    "SELECT user_order.id as id
    FROM user_order
    WHERE user_order.user_id = :user_id AND user_order.finished_at IS
NULL";
    $statement = $conn->prepare($query);
    $statement->bindParam(':user_id',$userId);
    $statement->execute();
    $userOrder = $statement->fetch();
    $userOrderId = $userOrder['id'];

    for($i=0;$i<count($booksToUpdate);$i++){
        $query =
        "UPDATE order_book
        SET order_book.quantity = :quantity
        WHERE order_book.order_id = :order_id AND order_book.book_id
= :book_id";

        $statement = $conn->prepare($query);
        $statement->bindParam(':quantity',$booksToUpdate[$i]->quantity);
        $statement->bindParam(':book_id',$booksToUpdate[$i]->id);
        $statement->bindParam(':order_id',$userOrderId);

        $statement->execute();
    }
}

function insertCartItems($userId,$booksToInsert){
    global $conn;

    $query =
    "SELECT user_order.id as id
    FROM user_order

```

```

        WHERE user_order.user_id = :user_id AND user_order.finished_at IS
NULL";
        $statement = $conn->prepare($query);
        $statement->bindParam(':user_id',$userId);
        $statement->execute();

        $userOrder = $statement->fetch();
        $userOrderId = $userOrder['id'];

        for($i=0;$i<count($booksToInsert);$i++){
            $query =
                "INSERT INTO order_book (order_id,book_id,quantity,unit_price)
                VALUES(:order_id,:book_id,:quantity,
                (SELECT book.price
                FROM book
                WHERE book.id = :book_id))";

            $statement = $conn->prepare($query);
            $statement->bindParam(":book_id",$booksToInsert[$i]->id);
            $statement->bindParam(":quantity",$booksToInsert[$i]->quantity);
            $statement->bindParam(":order_id",$userOrderId);

            $statement->execute();
        }
    }

function deleteUserCartItems($userId,$itemsToDelete){
    global $conn;

    $query =
        "SELECT user_order.id as id
        FROM user_order
        WHERE user_order.user_id = :user_id AND user_order.finished_at IS
NULL";
    $statement = $conn->prepare($query);
    $statement->bindParam(':user_id',$userId);
    $statement->execute();

    $userOrder = $statement->fetch();
    $userOrderId = $userOrder['id'];

    for($i=0;$i<count($itemsToDelete);$i++){
        $query =
            "DELETE
            FROM order_book
            WHERE order_book.order_id = :order_id AND order_book.book_id
            = :book_id";

```

```

    $statement = $conn->prepare($query);
    $statement->bindParam(":book_id",$itemsToDelete[$i]);
    $statement->bindParam(":order_id",$userOrderId);

    $statement->execute();
}
}

function getCartId($userId){
    global $conn;

    $query =
    "SELECT user_order.id as id
    FROM user_order
    WHERE user_order.user_id = :user_id AND user_order.finished_at IS
NULL";
    $statement = $conn->prepare($query);
    $statement->bindParam(':user_id',$userId);
    $statement->execute();

    $user_order = $statement->fetch();
    $userOrderId = $user_order['id'];

    return $userOrderId;
}
?>

```

Activate-category.php

```
<?php
    session_start();
    ob_start();
    header('Content-type: application/json');

    require_once "../../config/connection.php";
    require_once "../../global/functions.php";
    require_once "functions.php";

    if($_SERVER['REQUEST_METHOD']=='GET'){
        http_response_code(404);
        redirect("../index.php?page=404");
        exit;
    }

    $errorMessageArray = [];

    if(!isset($_SESSION['userId'])){
        http_response_code(401);
        $errorMessageArray[] = 'Access denied!';
        echo json_encode($errorMessageArray);
        exit;
    }

    if($_SESSION['userRoleName'] != 'admin'){
        http_response_code(403);
        $errorMessageArray[] = 'Access denied!';
        echo json_encode($errorMessageArray);
        exit;
    }

    $categoryId = $_POST['category-id'];

    try{
        activateCategory($categoryId);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
        exit;
    }
```

```
try{
    $allCategories = selectAllCategories();
}catch(PDOException $ex){
    http_response_code(500);
    $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
    echo json_encode($errorMessageArray);
    exit;
}

$message[] = $allCategories;
http_response_code(200);
echo json_encode($message);
ob_end_flush();
?>
```

Add-update-categories.php

```
<?php
    session_start();
    ob_start();
    header('Content-type: application/json');

    require_once "../../config/connection.php";
    require_once "../../global/functions.php";
    require_once "functions.php";

    if($_SERVER['REQUEST_METHOD']=='GET'){
        http_response_code(404);
        redirect("../index.php?page=404");
        exit;
    }

    $errorMessageArray = [];

    if(!isset($_SESSION['userId'])){
        http_response_code(401);
        $errorMessageArray[] = 'Access denied!';
        echo json_encode($errorMessageArray);
        exit;
    }

    if($_SESSION['userRoleName'] != 'admin'){
        http_response_code(403);
        $errorMessageArray[] = 'Access denied!';
        echo json_encode($errorMessageArray);
        exit;
    }

    $categoryId = $_POST['category-id'];
    $categoryName = $_POST['category-name'];
    $categoryType = $_POST['category-type'];
    $subcategoryParent = $_POST['select-category'];

    checkInputElementWithRegex('/^([A-ZŠĐĆŽ][a-zđćčž]{1}){2,}([A-ZŠĐĆŽ][a-zđćčž]{1}){2,}*/','$categoryName,'Incorrect
format',$errorMessageArray);

    if($errorMessageArray){
```

```

        http_response_code(400);
        echo json_encode($errorMessageArray);
        exit;
    }

    if($categoryType == 'category'){
        if($categoryId == '0'){
            try{
                addCategory($categoryName);
            }catch(PDOException $ex){
                http_response_code(500);
                $errorMessageArray[] = 'Server error: ' . $ex-
>getMessage();
                echo json_encode($errorMessageArray);
                exit;
            }
        }else{
            try{
                updateCategory($categoryId,$categoryName);
            }catch(PDOException $ex){
                http_response_code(500);
                $errorMessageArray[] = 'Server error: ' . $ex-
>getMessage();
                echo json_encode($errorMessageArray);
                exit;
            }
        }
    }

    if($categoryType == 'subcategory'){
        if($categoryId == '0'){
            try{
                addSubcategory($categoryName,$subcategoryParent);
            }catch(PDOException $ex){
                http_response_code(500);
                $errorMessageArray[] = 'Server error: ' . $ex-
>getMessage();
                echo json_encode($errorMessageArray);
                exit;
            }
        }else{
            try{
                updateSubcategory($categoryId,$categoryName,$subcategory
Parent);
            }catch(PDOException $ex){
                http_response_code(500);
                $errorMessageArray[] = 'Server error: ' . $ex-
>getMessage();
            }
        }
    }
}

```

```
        echo json_encode($errorMessageArray);
        exit;
    }
}

try{
    $allCategories = selectAllCategories();
}catch(PDOException $ex){
    http_response_code(500);
$errorMessageArray[] = 'Server error: ' . $ex->getMessage();
    echo json_encode($errorMessageArray);
    exit;
}

$message[] = $allCategories;
http_response_code(200);
echo json_encode($message);
ob_end_flush();
?>
```

Delete-category.php

```
<?php
    session_start();
    ob_start();
    header('Content-type: application/json');

    require_once "../../config/connection.php";
    require_once "../../global/functions.php";
    require_once "functions.php";

    if($_SERVER['REQUEST_METHOD']=='GET'){
        http_response_code(404);
        redirect("../index.php?page=404");
        exit;
    }

    $errorMessageArray = [];

    if(!isset($_SESSION['userId'])){
        http_response_code(401);
        $errorMessageArray[] = 'Access denied!';
        echo json_encode($errorMessageArray);
        exit;
    }

    if($_SESSION['userRoleName'] != 'admin'){
        http_response_code(403);
        $errorMessageArray[] = 'Access denied!';
        echo json_encode($errorMessageArray);
        exit;
    }

    $categoryId = $_POST['category-id'];

    try{
        deleteCategory($categoryId);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
        exit;
    }
```

```
try{
    $allCategories = selectAllCategories();
}catch(PDOException $ex){
    http_response_code(500);
    $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
    echo json_encode($errorMessageArray);
    exit;
}

$message[] = $allCategories;
http_response_code(200);
echo json_encode($message);
ob_end_flush();
?>
```

Get-active-categories.php

```
<?php
    session_start();
    ob_start();
    header('Content-type: application/json');

    require_once "../config/connection.php";
    require_once "../global/functions.php";
    require_once "../category/functions.php";

    if($_SERVER['REQUEST_METHOD']=='GET'){
        http_response_code(404);
        redirect("../index.php?page=404");
        exit;
    }

    $errorMessageArray = [];

    try{
        $allParentCategories = selectAllActiveCategories(null);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
        exit;
    }

    try{
        $allCategories = getCategories($allParentCategories);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
        exit;
    }

    http_response_code(200);
    echo json_encode($allCategories);
    ob_end_flush();
?>
```

Models > category > functions.php

```
<?php

function selectAllActiveCategories($parent){
    global $conn;

    if($parent){
        $query = 'SELECT * FROM category WHERE category_id = :parent
AND status = 1';
    }else{
        $query = 'SELECT * FROM category WHERE category_id IS NULL AND
status = 1';
    }

    $statement = $conn->prepare($query);

    if($parent){
        $statement->bindParam(':parent', $parent);
    }

    $statement->execute();

    $allCategories = $statement->fetchAll();

    return $allCategories;
}

function getCategories($allParentCategories){
    global $conn;

    $valuesNotIn = "0 ";
    $resultsArray = [];

    for($i=0;$i<count($allParentCategories);$i++){
        $query =
            "SELECT DISTINCT book.id, book.name,image.src, 1 AS
'category_name', 1 AS category_id
        FROM book
        JOIN book_category ON book_category.book_id = book.id
        JOIN publisher ON publisher.id = book.publisher_id
        JOIN image ON image.id = book.image_id
        WHERE book_category.category_id IN (SELECT child.id
                                            FROM book_category
                                            WHERE book_category.parent_id = :parent
                                            AND book_category.status = 1)
        ORDER BY category_name ASC";
        $statement = $conn->prepare($query);
        $statement->bindParam(':parent', $allParentCategories[$i]);
        $statement->execute();
        $results = $statement->fetchAll();
        $resultsArray = array_merge($resultsArray, $results);
    }
}

function getAllBooks($category_id){
    global $conn;

    $query = "
        SELECT book.id, book.name, image.src, publisher.name AS
'publisher_name', category_name
        FROM book
        JOIN book_category ON book_category.book_id = book.id
        JOIN publisher ON publisher.id = book.publisher_id
        JOIN image ON image.id = book.image_id
        WHERE book_category.category_id = :category_id
        ORDER BY book.name ASC";
    $statement = $conn->prepare($query);
    $statement->bindParam(':category_id', $category_id);
    $statement->execute();
    $books = $statement->fetchAll();
    return $books;
}
```

```

        FROM category parent
        JOIN category child ON
parent.id = child.category_id
                                WHERE parent.id
= :parent_id)
        AND book.id NOT IN ($valuesNotIn) AND book.status = 1 AND
publisher.status = 1
        GROUP BY book.id
        LIMIT 1";

$statement = $conn->prepare($query);
$statement-
>bindParam(':parent_id',$allParentCategories[$i]['id']);
$statement->execute();

$result = $statement->fetch();
$result['category_name'] = $allParentCategories[$i]['name'];
$result['category_id'] = $allParentCategories[$i]['id'];
$valuesNotIn .= ', ' . $result['id'];
$resultsArray[] = $result;
}

$filteredArray = [];
$isUnique = true;

foreach ($resultsArray as $item) {
    foreach($filteredArray as $cat){
        if($cat['category_id'] == $item['category_id']){
            $isUnique = false;
            break;
        }
    }

    if(!$isUnique){
        $isUnique = true;
        continue;
    }

    $filteredArray[] = $item;
}

return $filteredArray;
}

function selectUserPreferredCategories($userId){
    global $conn;
}

```

```

        $query = "SELECT category_id AS id FROM prefered_categories WHERE
user_id = :user_id";
        $statement = $conn->prepare($query);

        $statement->bindParam(':user_id', $userId);
        $statement->execute();

        $result = $statement->fetchAll();

        return $result;
    }

function displayCategoryArticle($category){

    $display = '<article class="article-category">
        <a href="index.php?page=shop&category-id='.
$category['category_id'] .'"></a>
        <p>' . $category['category_name'] . '</p>
        </article>';

    return $display;
}

function displayCategoriesMenu($parent,$checkId){

    $allCategories = selectAllActiveCategories($parent);

    $checked = $parent == $checkId ? "checked" : "";

    if(count($allCategories)){
        echo "<ul>";
        foreach($allCategories as $category){
            echo "<li class='book-category-li'><input type='checkbox'" .
$checked ." name='book-category-cb[]' class='book-category-cb' value='".
$category['id'] . "' id='cb-' . $category['id']. ''><label for='cb-
".$category['id'] ."'>". $category['name'] . "</label></li>";
            }
            echo "</ul>";
        }
    }

function selectRelatedCategoriesToABook($bookId){
    global $conn;

    $query = "SELECT parent.id, parent.name
    FROM category parent

```

```

        JOIN category child ON parent.id = child.category_id
        WHERE child.id IN (SELECT category.id
                            FROM book
                            JOIN image ON image.id = book.image_id
                            JOIN book_category ON book_category.book_id =
book.id
                            JOIN category ON category.id =
book_category.category_id
                            WHERE book.id = :book_id)";

$stmt = $conn->prepare($query);
$stmt->bindParam(':book_id', $bookId);

$stmt->execute();
$result = $stmt->fetchAll();

return $result;
}

function selectAllCategories(){
    global $conn;

    $query = 'SELECT c.id AS id, c.name AS name, p.name AS
parent,COUNT(book_category.book_id) AS book_count, c.status AS status,
c.created_at AS created_at, c.deleted_at AS deleted_at, c.category_id
AS category_id FROM category c LEFT JOIN category p ON c.category_id =
p.id LEFT JOIN book_category ON book_category.category_id = c.id GROUP
BY c.id';
    $stmt = $conn->prepare($query);

    $stmt->execute();

    $allCategorie = $stmt->fetchAll();

    for($i=0; $i<count($allCategorie);$i++){
        if($allCategorie[$i]['category_id']){
            continue;
        }

        $query = 'SELECT COUNT(DISTINCT book_category.book_id) AS count
FROM category
JOIN category child_category ON child_category.category_id =
category.id
JOIN book_category ON book_category.category_id =
child_category.id
WHERE category.id = :category_id';

$stmt = $conn->prepare($query);

```

```

    $statement->bindParam(':category_id', $allCategorie[$i]['id']);
    $statement->execute();
    $result = $statement->fetch(PDO::FETCH_OBJ);
    $allCategorie[$i]['book_count'] = $result->count;

}

$activeCategories = array();
$deletedCategories = array();

foreach ($allCategorie as $category) {
    if ($category['status'] == 1) {
        $activeCategories[] = $category;
    } elseif ($category['status'] == 0) {
        $deletedCategories[] = $category;
    }
}

$categoriesTable = new stdClass();
$categoriesTable->activeCategories = $activeCategories;
$categoriesTable->deletedCategories = $deletedCategories;

return $categoriesTable;
}

function
updateSubcategory($categoryId,$categoryName,$subcategoryParent){
    global $conn;

    $query = 'UPDATE category SET name = :category_name, category_id
= :subcategory_parent WHERE id = :category_id';
    $statement = $conn->prepare($query);

    $statement->bindParam(':category_name',$categoryName);
    $statement->bindParam(':category_id',$categoryId);
    $statement->bindParam(':subcategory_parent',$subcategoryParent);
    $statement->execute();
}

function addSubcategory($categoryName,$subcategoryParent){
    global $conn;

    $query = 'INSERT INTO category (name,status,category_id)
VALUES(:category_name,1,:category_id)';
    $statement = $conn->prepare($query);

    $statement->bindParam(':category_name',$categoryName);
    $statement->bindParam(':category_id',$subcategoryParent);
}

```

```

        $statement->execute();
    }

function addCategory($categoryName){
    global $conn;

    $query = 'INSERT INTO category (name,status)
VALUES(:category_name,1)';
    $statement = $conn->prepare($query);

    $statement->bindParam(':category_name',$categoryName);
    $statement->execute();
}

function updateCategory($categoryId,$categoryName){
    global $conn;

    $query = 'UPDATE category SET name = :category_name WHERE id
= :category_id';
    $statement = $conn->prepare($query);

    $statement->bindParam(':category_name',$categoryName);
    $statement->bindParam(':category_id',$categoryId);
    $statement->execute();
}

function deleteCategory($categoryId){
    global $conn;

    $query = 'UPDATE category SET status = 0, deleted_at = NOW() WHERE
id = :category_id';
    $statement = $conn->prepare($query);
    $statement->bindParam(':category_id',$categoryId);

    $statement->execute();

    $query2 = 'UPDATE category SET status = 0, deleted_at = NOW() WHERE
category_id = :category_id';
    $statement2 = $conn->prepare($query2);
    $statement2->bindParam(':category_id',$categoryId);

    $statement2->execute();
}

function activateCategory($categoryId){
    global $conn;

    $query = 'UPDATE category SET status = 1 WHERE id = :category_id';

```

```

$statement = $conn->prepare($query);
$statement->bindParam(':category_id',$categoryId);

$statement->execute();
}

function deleteUpdatedBookCategories($bookId,$deletedCategories){
    global $conn;

    for($i=0; $i<count($deletedCategories);$i++){
        $query = 'DELETE FROM book_category WHERE book_id = :book_id
AND category_id = :category_id';
        $statement = $conn->prepare($query);

        $statement->bindParam(':book_id',$bookId);
        $statement->bindParam(':category_id',$deletedCategories[$i]);

        $statement->execute();
    }
}

function addUpdatedBookCategories($bookId,$addedCategories){
    global $conn;

    for($i=0; $i<count($addedCategories);$i++){
        $query = 'INSERT INTO book_category (book_id, category_id)
VALUES (:book_id, :category_id)';
        $statement = $conn->prepare($query);

        $statement->bindParam(':book_id',$bookId);
        $statement->bindParam(':category_id',$addedCategories[$i]);

        $statement->execute();
    }
}
?>

```

Activate-delivery-type.php

```
<?php
    session_start();
    ob_start();
    header('Content-type: application/json');

    require_once "../../config/connection.php";
    require_once "../../global/functions.php";
    require_once "functions.php";

    if($_SERVER['REQUEST_METHOD']=='GET'){
        http_response_code(404);
        redirect("../index.php?page=404");
        exit;
    }

    $errorMessageArray = [];

    if(!isset($_SESSION['userId'])){
        http_response_code(401);
        $errorMessageArray[] = 'Access denied!';
        echo json_encode($errorMessageArray);
        exit;
    }

    if($_SESSION['userRoleName'] != 'admin'){
        http_response_code(403);
        $errorMessageArray[] = 'Access denied!';
        echo json_encode($errorMessageArray);
        exit;
    }

    $deliveryTypeId = $_POST['delivery-type-id'];

    try{
        activateDeletedDeliveryType($deliveryTypeId);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
        exit;
    }

    try{
```

```
$allDeliveryTypes = selectAllDeliveryTypes();  
}catch(PDOException $ex){  
    http_response_code(500);  
    $errorMessageArray[] = 'Server error: ' . $ex->getMessage();  
    echo json_encode($errorMessageArray);  
    exit;  
}  
  
$messsage[ ] = $allDeliveryTypes;  
http_response_code(200);  
echo json_encode($messsage);  
ob_end_flush();  
?>
```

Add-update-delivery-type.php

```
<?php
    session_start();
    ob_start();
    header('Content-type: application/json');

    require_once "../config/connection.php";
    require_once "../global/functions.php";
    require_once "functions.php";

    if($_SERVER['REQUEST_METHOD']=='GET'){
        http_response_code(404);
        redirect("../index.php?page=404");
        exit;
    }

    $errorMessageArray = [];

    if(!isset($_SESSION['userId'])){
        http_response_code(401);
        $errorMessageArray[] = 'Access denied!';
        echo json_encode($errorMessageArray);
        exit;
    }

    if($_SESSION['userRoleName'] != 'admin'){
        http_response_code(403);
        $errorMessageArray[] = 'Access denied!';
        echo json_encode($errorMessageArray);
        exit;
    }

    $deliveryTypeId = $_POST['delivery-type-id'];
    $deliveryTypeName = $_POST['delivery-type-name'];

    checkInputElementWithRegex('/^([A-ZŠĐĆŽ][a-zđćčž]{2,})( [A-ZŠĐĆŽ][a-zđćčž]{2,})*$/',$deliveryTypeName,'Incorrect
format',$errorMessageArray);

    if($errorMessageArray){
        http_response_code(400);
        echo json_encode($errorMessageArray);
        exit;
    }
```

```

if($deliveryTypeId == '0'){
    try{
        addDeliveryType($deliveryTypeName);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
        exit;
    }
}else{
    try{
        updateDeliveryType($deliveryTypeId,$deliveryTypeName);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
        exit;
    }
}

try{
    $allDeliveryTypes = selectAllDeliveryTypes();
}catch(PDOException $ex){
    http_response_code(500);
    $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
    echo json_encode($errorMessageArray);
    exit;
}

$message[] = $allDeliveryTypes;
http_response_code(200);
echo json_encode($message);
ob_end_flush();
?>

```

Delete-delivery-type.php

```
<?php
    session_start();
    ob_start();
    header('Content-type: application/json');

    require_once "../../config/connection.php";
    require_once "../../global/functions.php";
    require_once "functions.php";

    if($_SERVER['REQUEST_METHOD']=='GET'){
        http_response_code(404);
        redirect("../index.php?page=404");
        exit;
    }

    $errorMessageArray = [];

    if(!isset($_SESSION['userId'])){
        http_response_code(401);
        $errorMessageArray[] = 'Access denied!';
        echo json_encode($errorMessageArray);
        exit;
    }

    if($_SESSION['userRoleName'] != 'admin'){
        http_response_code(403);
        $errorMessageArray[] = 'Access denied!';
        echo json_encode($errorMessageArray);
        exit;
    }

    $deliveryTypeId = $_POST['delivery-type-id'];

    try{
        deleteDeliveryType($deliveryTypeId);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
        exit;
    }

    try{
```

```
$allDeliveryTypes = selectAllDeliveryTypes();  
}catch(PDOException $ex){  
    http_response_code(500);  
    $errorMessageArray[] = 'Server error: ' . $ex->getMessage();  
    echo json_encode($errorMessageArray);  
    exit;  
}  
  
$messsage[ ] = $allDeliveryTypes;  
http_response_code(200);  
echo json_encode($messsage);  
ob_end_flush();  
?>
```

Models > delivery > functions.php

```
<?php

function selectAllDeliveryTypes(){
    global $conn;

    $query =
        'SELECT d.id as id, d.name as name, d.created_at as created_at,
d.deleted_at as deleted_at, d.status as status,
        COUNT(user_order.id) as order_count
    FROM delivery_type d
    LEFT JOIN user_order ON user_order.delivery_type_id = d.id
    GROUP BY d.id';
    $statement = $conn->prepare($query);

    $statement->execute();

    $allDeliveryTypes = $statement->fetchAll();

    $activeDeliveryTypes = array();
    $deletedDeliveryTypes = array();

    foreach ($allDeliveryTypes as $deliveryType) {
        if ($deliveryType['status'] == 1) {
            $activeDeliveryTypes[] = $deliveryType;
        } elseif ($deliveryType['status'] == 0) {
            $deletedDeliveryTypes[] = $deliveryType;
        }
    }

    $deliveryTypeTable = new stdClass();
    $deliveryTypeTable->activeDeliveryTypes = $activeDeliveryTypes;
    $deliveryTypeTable->deletedDeliveryTypes = $deletedDeliveryTypes;

    return $deliveryTypeTable;
}

function addDeliveryType($deliveryTypeName){
    global $conn;

    $query = 'INSERT INTO delivery_type (name,status)
VALUES(:delivery_type_name,1)';
    $statement = $conn->prepare($query);
```

```

        $statement->bindParam(':delivery_type_name', $deliveryTypeName);
        $statement->execute();
    }

function updateDeliveryType($deliveryTypeId, $deliveryTypeName){
    global $conn;

    $query = 'UPDATE delivery_type SET name = :delivery_type_name WHERE
id = :delivery_type_id';
    $statement = $conn->prepare($query);

    $statement->bindParam(':delivery_type_name', $deliveryTypeName);
    $statement->bindParam(':delivery_type_id', $deliveryTypeId);
    $statement->execute();
}

function activateDeletedDeliveryType($deliveryTypeId){
    global $conn;

    $query = 'UPDATE delivery_type SET status=1 WHERE id
= :delivery_type_id';
    $statement = $conn->prepare($query);

    $statement->bindParam(':delivery_type_id', $deliveryTypeId);

    $statement->execute();
}

function deleteDeliveryType($deliveryTypeId){
    global $conn;

    $query = 'UPDATE delivery_type SET status=0, deleted_at = NOW()
WHERE id = :delivery_type_id';
    $statement = $conn->prepare($query);

    $statement->bindParam(':delivery_type_id', $deliveryTypeId);

    $statement->execute();
}

?>

```

Models > footer > functions.php

```
<?php
function allLinks($userRole){
    global $conn;

    $query = "SELECT l.name, l.href, lt.name AS 'target', ltype.name AS
'type', l.appearance FROM link as l
    INNER JOIN link_position as lp ON lp.link_id = l.id
    INNER JOIN position as p ON p.id = lp.position_id
    INNER JOIN link_role as lr ON lr.link_id = l.id
    INNER JOIN role as r ON r.id = lr.role_id
    INNER JOIN link_target as lt ON l.id_target = lt.id
    INNER JOIN link_type as ltype ON ltype.id = l.id_link_type
    WHERE p.name = 'footer' AND r.name = :role
    ORDER BY l.appearance ASC;";

    $statement = $conn->prepare($query);
    $statement->bindParam(":role",$userRole);

    $statement->execute();
    $links = $statement->fetchAll();

    $documentsLinks = array();
    $pageLinks = array();
    $socialMediaLinks = array();

    foreach($links as $link){
        if($link['type'] == 'document'){
            $documentsLinks[] = $link;
        }else if($link['type'] == 'link'){
            $pageLinks[] = $link;
        }else if($link['type'] == 'social media'){
            $socialMediaLinks[] = $link;
        }
    }

    $allLinks = new stdClass();
    $allLinks->documentsLinks = $documentsLinks;
    $allLinks->pageLinks = $pageLinks;
    $allLinks->socialMediaLinks = $socialMediaLinks;

    return $allLinks;
}
```

?>

Models > global > fucntions.php

```
<?php

function redirect($url){
    header("Location: ".$url);
}

function destroyAllCookies(){
    setcookie('isLoggedIn', false,time() - 36000,'/');
    setcookie('hasPreferedCategories', true,time() - 36000,'/');
    setcookie('cart', "",time() - 36000,'/');
}

function
checkInputElementWithRegex($regex,$element,$errorMessage,&$errorMessagesArray){
    if(preg_match($regex,$element)){
        return 0;
    }

    $errorMessagesArray[] = $errorMessage;
    return 1;
}

function recordUserVisit(){

    global $page;

    $open = fopen(LOG_FILE, "a");
    if($open){
        $date = date('d-m-Y H:i:s');
        fwrite($open,
 "{$page}\t{$date}\t{$_SERVER['REMOTE_ADDR']}\t\n");
        fclose($open);
    }
}
?>
```

Models > header > functions.php

```
<?php
require_once "config/connection.php";

function headerLinks($userRole){
    global $conn;

    $query = "SELECT l.name, l.href FROM link as l
INNER JOIN link_position as lp ON lp.link_id = l.id
INNER JOIN position as p ON p.id = lp.position_id
INNER JOIN link_role as lr ON lr.link_id = l.id
INNER JOIN role as r ON r.id = lr.role_id
WHERE p.name = 'header' AND r.name = :role
ORDER BY l.appearance ASC";

    $statement = $conn->prepare($query);
    $statement->bindParam(":role",$userRole);

    $statement->execute();
    $links = $statement->fetchAll();

    return $links;
}
?>
```

Models > home > functions.php

```
<?php

function getSurvey($name,$userId){
    global $conn;

    $query =
        "SELECT choice.id, choice.name as choice, survey.survey_text as
name, choice.src as src, survey.id AS survey_id
        FROM survey
        JOIN choice_survey ON choice_survey.survey_id = survey.id
        JOIN choice ON choice.id = choice_survey.choice_id
        WHERE survey.name = :name AND survey.status = 1 AND :user_id NOT IN
(
    SELECT user_id
    FROM survey_user
    JOIN survey ON survey.id = survey_user.survey_id
    WHERE survey.name = :survey_name
)";

    $statement = $conn->prepare($query);
    $statement->bindParam(':name',$name);
    $statement->bindParam(':user_id',$userId);
    $statement->bindParam(':survey_name',$name);

    $statement->execute();

    $result = $statement->fetchAll();

    return $result;
}

?>
```

Models > login > functions.php

```
<?php

function loginUser($username,$password){
    global $conn;

    $query =
        'SELECT user.id, user.username, IFNULL(user_order.id,0) as order_id,
        user.role_id, role.name, avatar.src, user.email
        FROM user
        JOIN role ON user.role_id = role.id LEFT JOIN avatar ON
        avatar.user_id = user.id
        LEFT JOIN user_order ON user_order.user_id = user.id
        WHERE username = :username
        AND status = 1 AND is_active = 1 AND password = :password AND
        is_banned = 0
        ORDER BY CASE WHEN user_order.finished_at IS NULL THEN 0 ELSE 1 END,
        user_order.finished_at ASC
        LIMIT 1';
    $statement = $conn->prepare($query);

    $statement->bindParam(":username",$username);
    $statement->bindParam(":password",$password);
    $statement->execute();
    $result = $statement->fetchAll();

    return $result;
}

function recordLoggedInUser($user){
    $open = fopen(LOGGED_ID_RECORD, "a");

    $username = $user[0]["username"];
    $email = $user[0]["email"];
    $role = $user[0]["name"];

    if($open){
        $date = date('d-m-Y H:i:s');
        fwrite($open, "{$username}\t{$email}\t{$role}\t{$date}\t\n");
        fclose($open);
    }
}
?>
```

login.php

```
<?php
    session_start();
    ob_start();
    header('Content-type: application/json');

    require_once "functions.php";
    require_once "../global/functions.php";
    require_once "../category/functions.php";
    require_once "../cart/functions.php";
    require_once "../../config/connection.php";
    require_once "../user/functions.php";

    if($_SERVER['REQUEST_METHOD']=='GET'){
        http_response_code(404);
        redirect("../index.php?page=404");
        exit;
    }

    $errorMessageArray = [];

    $username = $_POST['username-input'];
    $password = md5($_POST['password-input'] .
'tHisIsSaltedPassWoRd1234.');

    try{
        $user = loginUser($username,$password);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
        exit;
    }

    if(count($user) == 0){
        http_response_code(400);
        $errorMessageArray[] = 'Invalid username or password. Please
try again.';
        echo json_encode($errorMessageArray);
        exit;
    }

    if(count($user) > 1){
        http_response_code(400);
```

```

$errorMessageArray[] = 'An error occurred while processing your
request. Please try again later.' . count($user);
echo json_encode($errorMessageArray);
exit;
}

$_SESSION['userId'] = $user[0]['id'];
$_SESSION['username'] = $user[0]['username'];
$_SESSION['userRoleId'] = $user[0]['role_id'];
$_SESSION['userRoleName'] = $user[0]['name'];
$_SESSION['userAvatarSrc'] = $user[0]['src'];

setcookie('isLoggedIn', true, time() + 36000, '/');

try{
    $peferedCategories =
selectUserPreferredCategories($user[0]['id']);
}catch(PDOException $ex){
    http_response_code(500);
    $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
    echo json_encode($errorMessageArray);
    exit;
}

if(count($peferedCategories)){
    setcookie('hasPreferredCategories', true, time() + 36000, '/');
}

$orderId = $user[0]['order_id'];
if(!$orderId){
    try{
        $orderId = insertNewUserOrder($user[0]['id']);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
        exit;
    }
}

try{
    $userCart = selectUserCart($user[0]['id'], $orderId);
}catch(PDOException $ex){
    http_response_code(500);
    $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
    echo json_encode($errorMessageArray);
    exit;
}

```

```
setcookie('cart',json_encode($userCart),time() + 36000 ,"/");
recordLoggedInUser($user);
http_response_code(302);
redirect('../index.php?page=home');
ob_end_flush();
?>
```

Models > message > functions.php

```
<?php

function getUserMessages(){
    global $conn;

    $query =
        "SELECT message.title as subject, message.message_text as text,
message.created_at as date, user.email as email, user.username as
username
        FROM message
        JOIN user ON user.id = message.user_id";

    $statement = $conn->prepare($query);
    $statement->execute();

    $result = $statement->fetchAll();

    return $result;
}

function insertUserMessage($userId,$subject,$message){
    global $conn;

    $query = "INSERT INTO message(title,message_text,user_id)
VALUES(:subject,:message,:user_id)";
    $statement = $conn->prepare($query);
    $statement->bindParam(':subject',$subject);
    $statement->bindParam(' :message' ,$message);
    $statement->bindParam(' :user_id' , $userId);
    $statement->execute();
}

?>
```

Send-message.php

```
<?php
session_start();
ob_start();
header('Content-type: application/json');

require_once "../../config/connection.php";
require_once "../global/functions.php";
require_once "functions.php";

if($_SERVER['REQUEST_METHOD']=='GET'){
    http_response_code(404);
    redirect("../index.php?page=404");
    exit;
}

if(!isset($_SESSION['userId'])){
    http_response_code(401);
    $errorMessageArray[] = 'Access denied!';
    echo json_encode($errorMessageArray);
    exit;
}

$userId = $_SESSION['userId'];
$subject = $_POST['subject'];
$message = $_POST['message-input'];

if(strlen($subject)==0){
    $errorMessageArray[] = 'Subject can\'t be empty';
}

checkInputElementWithRegex('/(\s.*){4,}/',$message,'Message must have at least 5 words.',$errorMessageArray);

if($errorMessageArray){
    http_response_code(400);
    echo json_encode($errorMessageArray);
    exit;
}

try{
    insertUserMessage($userId,$subject,$message);
}catch(PDOException $ex){
    http_response_code(500);
}
```

```
$errorMessageArray[] = 'Server error: ' . $ex->getMessage();
echo json_encode($errorMessageArray);
exit;
}

$message = [];
http_response_code(200);
$message[] = 'Thank you! Your message has been sent successfully';
echo json_encode($message);
ob_end_flush();
?>
```

Models > order > functions.php

```
<?php

function getUserOrders(){
    global $conn;

    $query =
        "SELECT user.username, user.email,user_order.finished_at,
user_order.total_price,order_status.name
         FROM user_order
        JOIN user ON user.id = user_order.user_id
        JOIN order_status ON user_order.order_status_id = order_status.id";

    $statement = $conn->prepare($query);
    $statement->execute();

    $result = $statement->fetchAll();
    return $result;
}

function
finishOrder($cartId,$databaseCart,$addressId,$deliveryOptionId){
    global $conn;

    $totalPrice = 0;
    foreach($databaseCart as $book){
        $totalPrice += $book['price'] * $book['quantity'];
    }

    $query =
        "UPDATE user_order
         SET total_price = :total_price, finished_at = NOW(),
order_status_id = 1,
         address_id = :address_id, delivery_type_id = :delivery_type_id
        WHERE id = :cart_id";

    $statement = $conn->prepare($query);
    $statement->bindParam(":address_id",$addressId);
    $statement->bindParam(":delivery_type_id",$deliveryOptionId);
    $statement->bindParam(":total_price",$totalPrice);
    $statement->bindParam(":cart_id",$cartId);

    $statement->execute();
}
```

?>

order-logic.php

```

checkInputElementWithRegex('/^([A-ZŠĐĆČŽ][a-
zšđćčž]{2,}\s?)+$/',$state,'Incorrect state',$errorMessageArray);
checkInputElementWithRegex('/^\d{5,15}$/',$zipCode,'Incorrect zip
code',$errorMessageArray);
checkInputElementWithRegex('/^([A-ZŠĐĆČŽ][a-
zšđćčž]{2,}\s?)+$/',$country,'Incorrect country',$errorMessageArray);

try{
    $databaseCart = getCartUserInfo($userId);
}catch(PDOException $ex){
    http_response_code(500);
    $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
    echo json_encode($errorMessageArray);
    exit;
}

if(!count($databaseCart)){
    $errorMessageArray[] = 'Your cart is empty. Please add a book to
proceed with the purchase.';
}

if($errorMessageArray){
    http_response_code(400);
    echo json_encode($errorMessageArray);
    exit;
}

try{
    $addressIdObj =
doesAddressAlreadyExists($addressLine,$addressNumber,$city,$state,$zipC
ode,$country);
}catch(PDOException $ex){
    http_response_code(500);
    $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
    echo json_encode($errorMessageArray);
    exit;
}

if($addressIdObj){
    $addressId = $addressIdObj->id;
}else{
    try{
        $addressId =
insertNewAddress($addressLine,$addressNumber,$city,$state,$zipCode,$cou
ntry);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
    }
}

```

```

        echo json_encode($errorMessageArray);
        exit;
    }
}

try{
    $cartId = getCartId($userId);
}catch(PDOException $ex){
    http_response_code(500);
    $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
    echo json_encode($errorMessageArray);
    exit;
}

try{
    finishOrder($cartId,$databaseCart,$addressId,$deliveryOptionId);
}catch(PDOException $ex){
    http_response_code(500);
    $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
    echo json_encode($errorMessageArray);
    exit;
}

try{
    $orderId = insertNewUserOrder($userId);
}catch(PDOException $ex){
    http_response_code(500);
    $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
    echo json_encode($errorMessageArray);
    exit;
}

try{
    $userCart = selectUserCart($userId,$orderId);
}catch(PDOException $ex){
    http_response_code(500);
    $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
    echo json_encode($errorMessageArray);
    exit;
}

setcookie('cart',json_encode($userCart),time() + 36000 ,"/");

$messsage[] = 'Thank you for your order! Your purchase has been saved
and will be processed soon.';
http_response_code(200);
echo json_encode($messsage);
ob_end_flush();

```

?>

Activate-publisher.php

```
<?php
    session_start();
    ob_start();
    header('Content-type: application/json');

    require_once "../../config/connection.php";
    require_once "../../global/functions.php";
    require_once "functions.php";

    if($_SERVER['REQUEST_METHOD']=='GET'){
        http_response_code(404);
        redirect("../index.php?page=404");
        exit;
    }

    $errorMessageArray = [];

    if(!isset($_SESSION['userId'])){
        http_response_code(401);
        $errorMessageArray[] = 'Access denied!';
        echo json_encode($errorMessageArray);
        exit;
    }

    if($_SESSION['userRoleName'] != 'admin'){
        http_response_code(403);
        $errorMessageArray[] = 'Access denied!';
        echo json_encode($errorMessageArray);
        exit;
    }

    $publisherId = $_POST['publisher-id'];

    try{
        activateDeletedPublisher($publisherId);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
        exit;
    }

    try{
```

```
$allPublishers = selectAllPublishers();
}catch(PDOException $ex){
    http_response_code(500);
    $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
    echo json_encode($errorMessageArray);
    exit;
}

$message[] = $allPublishers;
http_response_code(200);
echo json_encode($message);
ob_end_flush();
?>
```

Add-update-publisher.php

```
<?php
    session_start();
    ob_start();
    header('Content-type: application/json');

    require_once "../config/connection.php";
    require_once "../global/functions.php";
    require_once "functions.php";

    if($_SERVER['REQUEST_METHOD']=='GET'){
        http_response_code(404);
        redirect("../index.php?page=404");
        exit;
    }

    $errorMessageArray = [];

    if(!isset($_SESSION['userId'])){
        http_response_code(401);
        $errorMessageArray[] = 'Access denied!';
        echo json_encode($errorMessageArray);
        exit;
    }

    if($_SESSION['userRoleName'] != 'admin'){
        http_response_code(403);
        $errorMessageArray[] = 'Access denied!';
        echo json_encode($errorMessageArray);
        exit;
    }

    $publisherId = $_POST['publisher-id'];
    $publisherName = $_POST['publisher-name'];

    checkInputElementWithRegex('/^([A-ZŠĐĆŽ][a-zđććž]{2,})([A-ZŠĐĆŽ][a-zđććž]{2,})*$/', $publisherName, 'Incorrect
format', $errorMessageArray);

    if($errorMessageArray){
        http_response_code(400);
        echo json_encode($errorMessageArray);
        exit;
    }
```

```

if($publisherId == '0'){
    try{
        addPublisher($publisherName);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
        exit;
    }
}else{
    try{
        updatePublisher($publisherId,$publisherName);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
        exit;
    }
}

try{
    $allPublishers = selectAllPublishers();
}catch(PDOException $ex){
    http_response_code(500);
    $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
    echo json_encode($errorMessageArray);
    exit;
}

$message[] = $allPublishers;
http_response_code(200);
echo json_encode($message);
ob_end_flush();
?>

```

Delete-publisher.php

```
<?php
    session_start();
    ob_start();
    header('Content-type: application/json');

    require_once "../config/connection.php";
    require_once "../global/functions.php";
    require_once "functions.php";

    if($_SERVER['REQUEST_METHOD']=='GET'){
        http_response_code(404);
        redirect("../index.php?page=404");
        exit;
    }

    $errorMessageArray = [];

    if(!isset($_SESSION['userId'])){
        http_response_code(401);
        $errorMessageArray[] = 'Access denied!';
        echo json_encode($errorMessageArray);
        exit;
    }

    if($_SESSION['userRoleName'] != 'admin'){
        http_response_code(403);
        $errorMessageArray[] = 'Access denied!';
        echo json_encode($errorMessageArray);
        exit;
    }

    $publisherId = $_POST['publisher-id'];

    try{
        deletePublisher($publisherId);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
        exit;
    }

    try{
```

```
$allPublishers = selectAllPublishers();
}catch(PDOException $ex){
    http_response_code(500);
    $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
    echo json_encode($errorMessageArray);
    exit;
}

$message[] = $allPublishers;
http_response_code(200);
echo json_encode($message);
ob_end_flush();
?>
```

Models > publisher > functions.php

```
<?php

function selectAllPublishers(){
    global $conn;

    $query =
        'SELECT publisher.id AS id, publisher.name AS name,
        COUNT(book.id) AS book_count,
        publisher.created_at AS created_at, publisher.deleted_at AS
        deleted_at, publisher.status AS status
        FROM publisher
        LEFT JOIN book ON book.publisher_id = publisher.id
        GROUP BY publisher.id';
    $statement = $conn->prepare($query);

    $statement->execute();

    $allPublishers = $statement->fetchAll();

    $activePublishers = array();
    $deletedPublishers = array();

    foreach ($allPublishers as $publisher) {
        if ($publisher['status'] == 1) {
            $activePublishers[] = $publisher;
        } elseif ($publisher['status'] == 0) {
            $deletedPublishers[] = $publisher;
        }
    }

    $publisherTable = new stdClass();
    $publisherTable->activePublishers = $activePublishers;
    $publisherTable->deletedPublishers = $deletedPublishers;

    return $publisherTable;
}

function addPublisher($publisherName){
    global $conn;

    $query = 'INSERT INTO publisher (name,status)
VALUES(:publisher_name,1)';
    $statement = $conn->prepare($query);
```

```

        $statement->bindParam(':publisher_name',$publisherName);
        $statement->execute();
    }

function updatePublisher($publisherId,$publisherName){
    global $conn;

    $query = 'UPDATE publisher SET name = :publisher_name WHERE id
= :publisher_id';
    $statement = $conn->prepare($query);

    $statement->bindParam(':publisher_name',$publisherName);
    $statement->bindParam(':publisher_id',$publisherId);
    $statement->execute();
}

function deletePublisher($publisherId){
    global $conn;

    $query = 'UPDATE publisher SET status=0, deleted_at = NOW() WHERE
id = :publisher_id';
    $statement = $conn->prepare($query);

    $statement->bindParam(':publisher_id',$publisherId);

    $statement->execute();
}

function activateDeletedPublisher($publisherId){
    global $conn;

    $query = 'UPDATE publisher SET status=1 WHERE id = :publisher_id';
    $statement = $conn->prepare($query);

    $statement->bindParam(':publisher_id',$publisherId);

    $statement->execute();
}

?>

```

shop.php

```
<?php
session_start();
header('Content-type: application/json');

require_once "../book/functions.php";
require_once "../../config/connection.php";

if($_SERVER['REQUEST_METHOD']=='GET'){
    http_response_code(404);
    redirect("../index.php?page=404");
    exit;
}

define('PAGE_BOOK_COUNT',20);

$filter = "";
$sort = "";

$activePage = 1;

if(isset($_POST['activePage'])){
    $activePage = $_POST['activePage'];
}

if(!empty($_POST['input-search'])){
    $filter .= " AND book.name LIKE '%" . $_POST['input-search'] . "%' ";
}

if(!empty($_POST['price-min'])) {
    $filter .= " AND book.price " . '>' . " " . $_POST['price-min'] . " ";
}

if(!empty($_POST['price-max'])){
    $filter .= " AND book.price " . '<' . " " . $_POST['price-max'] . " ";
}

if(!empty($_POST['book-category-cb'])){
    $categories = $_POST['book-category-cb'];
    $filter .= "AND book_category.category_id IN ( " . $categories[0];
    for($i=1;$i<count($categories);$i++){
        $filter .= ", " . $categories[$i];
    }
}
```

```

        $filter .= ")";
    }

switch($_POST['sort']){
    case 'popular':
        $sort .= "book_count DESC";
        break;
    case 'newest':
        $sort .= "book.release_date DESC";
        break;
    case 'price-asc':
        $sort .= "book.price ASC";
        break;
    case 'price-desc':
        $sort .= "book.price DESC";
        break;
    case 'name-asc':
        $sort .= "book.name ASC";
        break;
    case 'name-desc':
        $sort .= "book.name DESC";
        break;
}
}

try{
    $books = getBooksForShop($filter,$sort);
}catch(PDOException $ex){
    http_response_code(500);
    $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
    echo json_encode($errorMessageArray);
    exit;
}

$message[] = ceil(count($books)/PAGE_BOOK_COUNT);
$message[] = intval($activePage);

$start = (intval($activePage)-1)*PAGE_BOOK_COUNT;
$length = PAGE_BOOK_COUNT;
$message[] = array_slice($books, $start, $length);

http_response_code(200);
echo json_encode($message);
ob_end_flush();
?>

```

Activate-account.php

```
<?php
ob_start();
require_once "../../config/connection.php";
require_once "../global/functions.php";
require_once "functions.php";

if(!isset($_GET['token'])){
    http_response_code(400);
    redirect("../index.php?page=home");
}

$token = $_GET['token'];

activateNewUser($token);

redirect('../index.php?page=login');
ob_end_flush();
?>
```

Models > signup > functions.php

```
<?php

function generateToken() {
    $bytes = random_bytes(10);
    $token = bin2hex($bytes);
    return $token;
}

function
insertNewUser($firstName,$lastName,$username,$password,$email,$phone,$t
oken,$addressId,$role){
    global $conn;

    $query = 'INSERT INTO user (first_name, last_name, username,
password, email, phone, token, address_id, role_id) VALUES
(:first_name, :last_name, :username, :password, :email, :phone, :token,
:address_id, :role)';
    $statement = $conn->prepare($query);

    $statement->bindParam(':first_name', $firstName);
    $statement->bindParam(':last_name', $lastName);
    $statement->bindParam(':username', $username);
    $statement->bindParam(':password', $password);
    $statement->bindParam(':email', $email);
    $statement->bindParam(':phone', $phone);
    $statement->bindParam(':token', $token);
    $statement->bindParam(':address_id', $addressId);
    $statement->bindParam(':role', $role);

    $statement->execute();

    return $conn->lastInsertId();
}

function send_activation_email($toEmail, $activationLink) {
    $subject = "Activate Your Account";
    $message = '
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Activate Your Account</title>
    <style type="text/css">
```

```

body {
    font-family: Arial, sans-serif;
    font-size: 14px;
    line-height: 1.5;
    color: #333;
}
h1 {
    font-size: 20px;
    font-weight: bold;
    margin-top: 0;
    margin-bottom: 20px;
}
p {
    margin-bottom: 20px;
}
a {
    color: #fff !important;
    text-decoration: none;
}
a:hover {
    text-decoration: underline;
}
.button {
    display: inline-block;
    padding: 8px 16px;
    background-color: #0066cc;
    color: #fff;
    border-radius: 4px;
    text-decoration: none;
}
.button:hover {
    background-color: #0052a3;
}
</style>
</head>
<body>
    <h1>Activate Your Account</h1>
    <p>Hi,</p>
    <p>Thanks for creating an account with us! Please activate your account by clicking on the following link:</p>
        <p><a href="'.\$activationLink . '" class="button">Activate Your Account</a></p>
        <p>If you have any questions or concerns, don't hesitate to contact us.</p>
        <p>Best regards,<br>Readily</p>
</body>
</html>
';

```

```

$headers = 'MIME-Version: 1.0' . "\r\n";
$headers .= 'Content-type: text/html; charset=utf-8' . "\r\n";
$headers .= 'From: Readily <pp5104133@gmail.com>' . "\r\n";
$headers .= "Reply-To: pp5104133@gmail.com\r\n";

if(mail($toEmail, $subject, $message, $headers)) {
    return true;
}
$lastError = error_get_last();
echo "Error: " . $lastError['message'];

return false;
}

function activateNewUser($token){
    global $conn;

    $query = 'UPDATE user SET status = 1 WHERE token = :token';
    $statement = $conn->prepare($query);

    $statement->bindParam(':token', $token);
    $statement->execute();
}
?>

```

signup.php

```
<?php
    ob_start();
    header('Content-type: application/json');

    require_once "../config/connection.php";
    require_once "../global/functions.php";
    require_once "../user/functions.php";
    require_once "functions.php";

    if($_SERVER['REQUEST_METHOD']=='GET'){
        http_response_code(404);
        redirect("../index.php?page=404");
        exit;
    }

    $avatarName = $_FILES['user-avatar']['name'];
    $avatarTmpName = $_FILES['user-avatar']['tmp_name'];
    $avatarSize = $_FILES['user-avatar']['size'];
    $avatarType = pathinfo($_FILES['user-avatar']['name'],
PATHINFO_EXTENSION);

    $firstName = $_POST['first-name-input'];
    $lastName = $_POST['last-name-input'];
    $username = $_POST['username-input'];
    $password = md5($_POST['password-input'] .
'tHisIsSaltedPassWoRd1234.');
    $email = $_POST['email-input'];
    $phone = $_POST['phone-input'];
    $role = $_POST['role-input'];
    $biography = $_POST['biography-input'];

    $addressLine = $_POST['address-line-input'];
    $addressNumber = $_POST['number-input'];
    $city = $_POST['city-input'];
    $state = $_POST['state-input'];
    $zipCode = $_POST['zip-code-input'];
    $country = $_POST['country-input'];

    $hasAvatar = false;
    $hasAddress = false;

    $addressId = null;
```

```

$errorMessageArray = [];

if(!empty($avatarName)){
    checkAvatar($avatarSize,$avatarType,$errorMessageArray);

    $hasAvatar = true;
}

checkInputElementWithRegex('/^([A-ZŠĐĆŽ][a-zđčćž]{2,})([A-ZŠĐĆŽ][a-zđčćž]{2,})*$/', $firstName, 'Incorrect first name format (ex. Jhon)', $errorMessageArray);
checkInputElementWithRegex('/^([A-ZŠĐĆŽ][a-zđčćž]{2,})*$/', $lastName, 'Incorrect Last name format (ex. Smith)', $errorMessageArray);
checkInputElementWithRegex('/^([a-zA-Z0-9.šđčćčŠĐŽ\(\)\/\-\_]{5,})$/', $username, 'Your username must be at least 5 characters long and can only contain letters, numbers, periods, parentheses, forward slashes, hyphens, and underscores.', $errorMessageArray);
checkInputElementWithRegex('/^([a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,})$/', $email, 'Incorrect email format (ex. jhonsmith@gmail.com)', $errorMessageArray);
checkInputElementWithRegex('/^\d{5,15}$/', $phone, 'Incorrect phone format (ex. 0611234567)', $errorMessageArray);

if($role != '2' && $role != '3'){
    $errorMessage[] = "Role is not valid";
} else if($role == '3'){
    checkInputElementWithRegex('/(\s.*){4,}/', $biography, 'Biography must have at least 5 words.', $errorMessageArray);
}

if(!empty($addressLine) || !empty($addressNumber) || !empty($city) || !empty($state) || !empty($zipCode) || !empty($country)){
    checkInputElementWithRegex('/^([a-zA-Z0-9\s\.\-\_#\]\+\$/)', $addressLine, 'Incorrect address line', $errorMessageArray);
    checkInputElementWithRegex('/^\d+[a-zA-Z]$/', $addressNumber, 'Incorrect address number', $errorMessageArray);
    checkInputElementWithRegex('/^([A-ZŠĐĆŽ][a-zA-Zđčćž]{2,}\s?)+$/', $city, 'Incorrect city', $errorMessageArray);
    checkInputElementWithRegex('/^([A-ZŠĐĆŽ][a-zA-Zđčćž]{2,}\s?)+$/', $state, 'Incorrect state', $errorMessageArray);
    checkInputElementWithRegex('/^\d{5,15}$/', $zipCode, 'Incorrect zip code', $errorMessageArray);
    checkInputElementWithRegex('/^([A-ZŠĐĆŽ][a-zA-Zđčćž]{2,}\s?)+$/', $country, 'Incorrect country', $errorMessageArray);

    $hasAddress = true;
}

```

```

if($errorMessageArray){
    http_response_code(400);
    echo json_encode($errorMessageArray);
    exit;
}

if($hasAddress){
    try{
        $addressIdObj =
doesAddressAlreadyExists($addressLine,$addressNumber,$city,$state,$zipC
ode,$country);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
        exit;
    }

    if($addressIdObj){
        $addressId = $addressIdObj->id;
    }else{
        try{
            $addressId =
insertNewAddress($addressLine,$addressNumber,$city,$state,$zipCode,$cou
ntry);
        }catch(PDOException $ex){
            http_response_code(500);
            $errorMessageArray[] = 'Server error: ' . $ex-
>getMessage();
            echo json_encode($errorMessageArray);
            exit;
        }
    }
}

$token = generateToken();

try{
    $userId =
insertNewUser($firstName,$lastName,$username,$password,$email,$phone,$t
oken,$addressId,$role);
}catch(PDOException $ex){
    http_response_code(500);
    $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
    echo json_encode($errorMessageArray);
    exit;
}

```

```

if($hasAvatar){

    try{
        $idAvatar = insertNewAvatarRow($userId);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
        exit;
    }

    $uploadDir = '.../..../assets/images/avatars/';
    $filePath = $uploadDir . $idAvatar . '.' . $avatarType;
    $result = move_uploaded_file($avatarTmpName,$filePath);

    if(!$result){
        http_response_code(500);
        $errorMessageArray[] = 'Server error.';
        echo json_encode($errorMessageArray);
        exit;
    }

    try{
        updateAvatarRowName($idAvatar,$avatarType);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
        exit;
    }
}

if($role == '3'){

    try{
        insertNewBiography($biography,$userId);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
        exit;
    }
}

$activationLink =
"http://localhost/Readilyphp/models/signup/activate-
account.php?token=" . $token;
if(!send_activation_email($email, $activationLink)) {
    $errorMessageArray[] = "Failed to send activation email.";
}

```

```
}

http_response_code(200);
$message[] = 'Hi! Your account has been created successfully.
Please check your email for activation instructions. Thank you!';
echo json_encode($message);
ob_end_flush();
?>
```

Activate-survey.php

```
<?php
    session_start();
    ob_start();
    header('Content-type: application/json');

    require_once "../config/connection.php";
    require_once "../global/functions.php";
    require_once "functions.php";

    if($_SERVER['REQUEST_METHOD']=='GET'){
        http_response_code(404);
        redirect("../index.php?page=404");
        exit;
    }

    $errorMessageArray = [];

    if(!isset($_SESSION['userId']) || $_SESSION['userRoleName'] != 'admin'){
        http_response_code(400);
        $errorMessageArray[] = 'Access denied!';
        echo json_encode($errorMessageArray);
        exit;
    }

    $surveyId = $_POST['survey-id'];

    try{
        activateDeletedSurvey($surveyId);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
        exit;
    }

    try{
        $allSurveys = selectAllSurveys();
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
        exit;
    }
```

```
}

$message[] = $allSurveys;
http_response_code(200);
echo json_encode($message);
ob_end_flush();
?>
```

Delete-survey.php

```
<?php
    session_start();
    ob_start();
    header('Content-type: application/json');

    require_once "../config/connection.php";
    require_once "../global/functions.php";
    require_once "functions.php";

    if($_SERVER['REQUEST_METHOD']=='GET'){
        http_response_code(404);
        redirect("../index.php?page=404");
        exit;
    }

    $errorMessageArray = [];

    if(!isset($_SESSION['userId'])){
        http_response_code(401);
        $errorMessageArray[] = 'Access denied!';
        echo json_encode($errorMessageArray);
        exit;
    }

    if($_SESSION['userRoleName'] != 'admin'){
        http_response_code(403);
        $errorMessageArray[] = 'Access denied!';
        echo json_encode($errorMessageArray);
        exit;
    }

    $surveyId = $_POST['survey-id'];

    try{
        deleteSurvey($surveyId);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
        exit;
    }

    try{
```

```
$allSurveys = selectAllSurveys();
}catch(PDOException $ex){
    http_response_code(500);
    $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
    echo json_encode($errorMessageArray);
    exit;
}

$message[] = $allSurveys;
http_response_code(200);
echo json_encode($message);
ob_end_flush();
?>
```

Models > survey > functions.php

```
<?php

function selectAllSurveys(){
    global $conn;

    $query =
        "SELECT survey.name as survey,survey.status as status,
        survey.id as survey_id,survey.finished_at as finished_at,
        survey.created_at as created_at,
        choice.name as choice,
        COUNT(survey_user.user_id) as votes
        FROM survey_user
        JOIN survey ON survey.id = survey_user.survey_id
        JOIN choice ON choice.id = survey_user.choice_id
        GROUP BY survey_user.survey_id, survey_user.choice_id
        ORDER BY survey.id ASC, choice.id ASC";

    $statement = $conn->prepare($query);
    $statement->execute();

    $result = $statement->fetchAll();

    $activeSurveys = array();
    $deletedSurveys = array();

    foreach ($result as $survey) {
        if ($survey['status'] == 1) {
            $activeSurveys[] = $survey;
        } elseif ($survey['status'] == 0) {
            $deletedSurveys[] = $survey;
        }
    }

    $surveyResults = [];
    foreach($activeSurveys as $survey){
        $surveyName = $survey['survey'];
        $surveyId = $survey['survey_id'];
        $optionName = $survey['choice'];
        $optionValue = $survey['votes'];
        $createdAt = $survey['created_at'];

        if (!isset($surveyResults[$surveyName])) {
            $surveyResults[$surveyName] = array(
                'name' => $surveyName,
                'id' => $surveyId,
                'options' => [
                    'name' => $optionName,
                    'value' => $optionValue
                ],
                'createdAt' => $createdAt
            );
        } else {
            $surveyResults[$surveyName]['options'][$optionName] = $optionValue;
        }
    }
}
```

```

        'name' => $surveyName,
        'id' => $surveyId,
        'created_at'=> $createdAt,
        'options' => array(),
    );
}

$surveyResults[$surveyName]['options'][$optionName] =
$optionValue;
}

unset($activeSurveys);
$activeSurveys = $surveyResults;

unset($surveyResults);
$surveyResults = [];

foreach($deletedSurveys as $survey){
    $surveyName = $survey['survey'];
    $surveyId = $survey['survey_id'];
    $optionName = $survey['choice'];
    $optionValue = $survey['votes'];
    $finishedAt = $survey['finished_at'];

    if (!isset($surveyResults[$surveyName])) {
        $surveyResults[$surveyName] = array(
            'name' => $surveyName,
            'id' => $surveyId,
            'finished_at'=> $finishedAt,
            'options' => array(),
        );
    }

    $surveyResults[$surveyName]['options'][$optionName] =
$optionValue;
}

unset($deletedSurveys);

$deletedSurveys = $surveyResults;

$surveyTable = new stdClass();
$surveyTable->activeSurveys = $activeSurveys;
$surveyTable->deletedSurveys = $deletedSurveys;

return $surveyTable;
}

```

```

function deleteSurvey($surveyId){
    global $conn;

    $query = 'UPDATE survey SET status=0, finished_at = NOW() WHERE id
= :survey_id';
    $statement = $conn->prepare($query);

    $statement->bindParam(':survey_id',$surveyId);

    $statement->execute();
}

function activateDeletedSurvey($surveyId){
    global $conn;

    $query = 'UPDATE survey SET status=1 WHERE id = :survey_id';
    $statement = $conn->prepare($query);

    $statement->bindParam(':survey_id',$surveyId);

    $statement->execute();
}

function
insertSurveyWebsiteExperienceData($surveyId,$optionId,$userId){
    global $conn;

    $query = "INSERT INTO survey_user (survey_id, user_id, choice_id)
VALUES (:survey_id, :user_id, :choice_id)";

    $statement = $conn->prepare($query);
    $statement->bindParam(':survey_id',$surveyId);
    $statement->bindParam(':user_id',$userId);
    $statement->bindParam(':choice_id',$optionId);

    $statement->execute();
}
?>

```

Insert-survey-webisite-experience-data.php

```
<?php
    session_start();
    ob_start();
    header('Content-type: application/json');

    require_once "../config/connection.php";
    require_once "../global/functions.php";
    require_once "../user/functions.php";
    require_once "functions.php";

    if($_SERVER['REQUEST_METHOD']=='GET'){
        http_response_code(404);
        redirect("../index.php?page=404");
        exit;
    }

    $errorMessageArray = [];

    if(!isset($_SESSION['userId'])){
        http_response_code(401);
        $errorMessageArray[] = 'Access denied!';
        echo json_encode($errorMessageArray);
        exit;
    }

    $optionId = $_POST['survey-choice'];
    $surveyId = $_POST['survey-id'];
    $userId = $_SESSION['userId'];

    try{
        insertSurveyWebsiteExperienceData($surveyId,$optionId,$userId);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
        exit;
    }

    http_response_code(200);
    $message[] = 'Survey information saved successfully. Thank you for
your participation!';
    echo json_encode($message);
    ob_end_flush();
```

?>

Activate-user.php

```
<?php
    session_start();
    ob_start();
    header('Content-type: application/json');

    require_once "../config/connection.php";
    require_once "../global/functions.php";
    require_once "functions.php";

    $errorMessageArray = [];

    if($_SERVER['REQUEST_METHOD']=='GET'){
        http_response_code(404);
        redirect("../index.php?page=404");
        exit;
    }

    if(!isset($_SESSION['userId'])){
        http_response_code(401);
        $errorMessageArray[] = 'Access denied!';
        echo json_encode($errorMessageArray);
        exit;
    }

    if($_SESSION['userRoleName'] != 'admin'){
        http_response_code(403);
        $errorMessageArray[] = 'Access denied!';
        echo json_encode($errorMessageArray);
        exit;
    }

    $userId = $_POST['userId'];

    try{
        activateDeletedUser($userId);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
        exit;
    }

    try{
```

```
$allUsers = selectAllUsers();
}catch(PDOException $ex){
    http_response_code(500);
    $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
    echo json_encode($errorMessageArray);
    exit;
}

$messsage[ ] = $allUsers;
http_response_code(200);
echo json_encode($messsage);
ob_end_flush();
?>
```

Ban-user.php

```
<?php
    session_start();
    ob_start();
    header('Content-type: application/json');

    require_once "../config/connection.php";
    require_once "../global/functions.php";
    require_once "functions.php";

    $errorMessageArray = [];

    if($_SERVER['REQUEST_METHOD']=='GET'){
        http_response_code(404);
        redirect("../index.php?page=404");
        exit;
    }

    if(!isset($_SESSION['userId'])){
        http_response_code(401);
        $errorMessageArray[] = 'Access denied!';
        echo json_encode($errorMessageArray);
        exit;
    }

    if($_SESSION['userRoleName'] != 'admin'){
        http_response_code(403);
        $errorMessageArray[] = 'Access denied!';
        echo json_encode($errorMessageArray);
        exit;
    }

    $userId = $_POST['userId'];

    try{
        banUser($userId);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
        exit;
    }

    try{
```

```
$allUsers = selectAllUsers();
}catch(PDOException $ex){
    http_response_code(500);
    $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
    echo json_encode($errorMessageArray);
    exit;
}

$messsage[ ] = $allUsers;
http_response_code(200);
echo json_encode($messsage);
ob_end_flush();
?>
```

Delete-user.php

```
<?php
    session_start();
    ob_start();
    header('Content-type: application/json');

    require_once "../config/connection.php";
    require_once "../global/functions.php";
    require_once "functions.php";

    $errorMessageArray = [];

    if($_SERVER['REQUEST_METHOD']=='GET'){
        http_response_code(404);
        redirect("../index.php?page=404");
        exit;
    }

    if(!isset($_SESSION['userId'])){
        http_response_code(401);
        $errorMessageArray[] = 'Access denied!';
        echo json_encode($errorMessageArray);
        exit;
    }

    $userId = $_POST['userId'];

    try{
        deleteUser($userId);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
        exit;
    }

    if($_SESSION['userRoleName'] != 'admin'){
        destroyAllCookies();
        session_destroy();
    }

    try{
        $allUsers = selectAllUsers();
    }catch(PDOException $ex){
```

```
http_response_code(500);
$errorMessageArray[] = 'Server error: ' . $ex->getMessage();
echo json_encode($errorMessageArray);
exit;
}

$message[] = $allUsers;
http_response_code(200);
echo json_encode($message);
ob_end_flush();
?>
```

Models > user > functions.php

```
<?php

function selectAllUsers(){

    global $conn;

    $query = 'SELECT user.id AS id, user.username AS username,
    user.email AS email, user.created_at AS created_at,
    user.deleted_at AS deleted_at, user.banned_at AS banned_at,
    user.is_active AS is_active, user.is_banned AS is_banned,
    role.name AS name
    FROM user JOIN role ON role.id = user.role_id';
    $statement = $conn->prepare($query);

    $statement->execute();

    $allUsers = $statement->fetchAll();

    $activeUsers = array();
    $deletedUsers = array();
    $bannedUsers = array();

    foreach ($allUsers as $user) {
        if ($user['is_active'] == 1) {
            $activeUsers[] = $user;
        } elseif ($user['is_active'] == 0 && $user['is_banned'] == 0) {
            $deletedUsers[] = $user;
        } elseif ($user['is_banned'] == 1) {
            $bannedUsers[] = $user;
        }
    }

    $userTable = new stdClass();
    $userTable->activeUsers = $activeUsers;
    $userTable->deletedUsers = $deletedUsers;
    $userTable->bannedUsers = $bannedUsers;

    return $userTable;
}

function getUserInformation($userId){
    global $conn;
```

```

$query =
    'SELECT user.id AS id, user.first_name AS first_name,
user.last_name AS last_name, user.username AS username,
    user.email AS email, user.phone AS phone,avatar.id AS avatar_id,
avatar.src AS avatar_src,role.id AS role_id, role.name AS role_name,
    biography.biography_text AS biography_text, address.address_line AS
address_line,
    address.address_number AS address_number, address.city AS city,
address.state AS state,
    address.zip_code AS zip_code, address.country AS country
    FROM user JOIN role ON user.role_id = role.id LEFT JOIN avatar ON
avatar.user_id = user.id
    LEFT JOIN biography ON user.id = biography.user_id LEFT JOIN
address ON address.id = user.address_id
    WHERE user.id = :user_id';
$stmt = $conn->prepare($query);

$stmt->bindParam(':user_id', $userId);
$stmt->execute();

$result = $stmt->fetch(PDO::FETCH_OBJ);

return $result;
}

function setUserInformationInCookie(){
    global $avatarId;
    global $avatarSrc;

    global $userId;
    global $firstName;
    global $lastName;
    global $username;
    global $email;
    global $phone;
    global $roleId;
    global $roleName;
    global $biographyText;

    global $addressLine;
    global $addressNumber;
    global $city;
    global $state;
    global $zipCode;
    global $country;

    setcookie("avatarId", $avatarId);
    setcookie("avatarSrc", $avatarSrc);
}

```

```

setcookie("userId", $userId);
setcookie("firstName", $firstName);
setcookie("lastName", $lastName);
setcookie("username", $username);
setcookie("email", $email);
setcookie("phone", $phone);
setcookie("roleId", $roleId);

if($roleName == 'writer'){
    setcookie("biographyText", $biographyText);
}

setcookie("addressLine", $addressLine);
setcookie("addressNumber", $addressNumber);
setcookie("city", $city);
setcookie("state", $state);
setcookie("zipCode", $zipCode);
setcookie("country", $country);

}

function deleteUser($userId){
    global $conn;

    $query = 'UPDATE user SET is_active = 0, deleted_at = NOW() WHERE
id = :user_id';
    $statement = $conn->prepare($query);

    $statement->bindParam(':user_id', $userId);

    $statement->execute();
}

function activateDeletedUser($userId){
    global $conn;

    $query = 'UPDATE user SET is_active = 1 WHERE id = :user_id';
    $statement = $conn->prepare($query);

    $statement->bindParam(':user_id', $userId);

    $statement->execute();
}

function unbanUser($userId){
    global $conn;

```

```

    $query = 'UPDATE user SET is_banned = 0,is_active=1, banned_at =
NOW() WHERE id = :user_id';
    $statement = $conn->prepare($query);

    $statement->bindParam(':user_id',$userId);

    $statement->execute();
}

function banUser($userId){
    global $conn;

    $query = 'UPDATE user SET is_banned = 1,is_active=0, banned_at =
NOW() WHERE id = :user_id';
    $statement = $conn->prepare($query);

    $statement->bindParam(':user_id',$userId);

    $statement->execute();
}

function checkAvatar($size,$type,&$errorMessagesArray){
    $allowedTypes = array('jpg', 'jpeg', 'png');

    if(!in_array($type, $allowedTypes)) {
        $errorMessagesArray[] = 'File type not allowed. Only JPG, JPEG,
and PNG files are allowed.';
        return;
    }

    if($size > 700 * 1024) {
        $errorMessagesArray[] = 'File size exceeds 700KB limit.';
        return;
    }
}

function doesAvatarAlreadyExists($userId){
    global $conn;

    $query = 'SELECT id, src FROM avatar WHERE user_id = :user_id';
    $statement = $conn->prepare($query);

    $statement->bindParam(':user_id',$userId);
    $statement->execute();

    return $statement->fetch(PDO::FETCH_OBJ);
}

```

```

function
doesAddressAlreadyExists($addressLine,$addressNumber,$city,$state,$zipC
ode,$country){
    global $conn;

        $query = 'SELECT id FROM address WHERE address_line = :address_line
AND address_number = :address_number
        AND city = :city AND state = :state AND zip_code = :zip_code AND
country = :country';
        $statement = $conn->prepare($query);

        $statement->bindParam(":address_line",$addressLine);
        $statement->bindParam(":address_number",$addressNumber);
        $statement->bindParam(":city",$city);
        $statement->bindParam(":state",$state);
        $statement->bindParam(":zip_code",$zipCode);
        $statement->bindParam(":country",$country);

        $statement->execute();
        $result = $statement->fetch(PDO::FETCH_OBJ);

        return $result;
}

function
insertNewAddress($addressLine,$addressNumber,$city,$state,$zipCode,$cou
ntry){
    global $conn;

        $query = 'INSERT INTO address (address_line, address_number, city,
state, zip_code, country) VALUES
(:address_line, :address_number, :city, :state, :zip_code, :country)';
        $statement = $conn->prepare($query);

        $statement->bindParam(':address_line', $addressLine);
        $statement->bindParam(':address_number', $addressNumber);
        $statement->bindParam(':city', $city);
        $statement->bindParam(':state', $state);
        $statement->bindParam(':zip_code', $zipCode);
        $statement->bindParam(':country', $country);

        $statement->execute();

        return $conn->lastInsertId();;
}

function updateAvatarRowName($avatarId,$avatarType){
    global $conn;

```

```

$src = $avatarId . '.' . $avatarType;

$query = 'UPDATE avatar SET src = :src WHERE id = :avatar_id';
$stmt = $conn->prepare($query);

$stmt->bindParam(":src",$src);
$stmt->bindParam(":avatar_id",$avatarId);
$stmt->execute();
}

function insertNewBiography($biography,$userId){
    global $conn;

    $query = 'INSERT INTO biography (biography_text, user_id) VALUES (:biography, :user_id)';
    $stmt = $conn->prepare($query);

    $stmt->bindParam(':biography', $biography);
    $stmt->bindParam(':user_id', $userId);

    $stmt->execute();
}

function insertNewAvatarRow($userId){
    global $conn;

    $query = "INSERT INTO avatar (src,user_id) VALUES (null,:user_id)";
    $stmt = $conn->prepare($query);

    $stmt->bindParam(":user_id",$userId);
    $stmt->execute();

    $lastInsertId = $conn->lastInsertId();

    return $lastInsertId;
}

function
updateUserInformation($userId,$firstName,$lastName,$username,$email,$ph
one,$addressId){
    global $conn;

    $query = 'UPDATE user SET first_name = :first_name, last_name
= :last_name, username = :username,
    email = :email, phone = :phone, address_id = :address_id WHERE id
= :user_id';
    $stmt = $conn->prepare($query);
}

```

```

    $statement->bindParam(':first_name', $firstName);
    $statement->bindParam(':last_name', $lastName);
    $statement->bindParam(':username', $username);
    $statement->bindParam(':email', $email);
    $statement->bindParam(':phone', $phone);
    $statement->bindParam(':address_id', $addressId);
    $statement->bindParam(':user_id', $userId);

    $statement->execute();
}

function updateUserBiography($userId,$biographyText){
    global $conn;

    $query = 'UPDATE biography SET biography_text = :biography_text
WHERE user_id = :user_id';
    $statement = $conn->prepare($query);

    $statement->bindParam(':biography_text', $biographyText);
    $statement->bindParam(':user_id', $userId);

    $statement->execute();
}

function insertUsersPreferedCategories($preferedCategories,$userId){
    global $conn;

    for($i=0;$i<count($preferedCategories);$i++){
        $query = "INSERT INTO prefered_categories (user_id, category_id)
VALUES (:user_id,:category_id)";
        $statement = $conn->prepare($query);

        $statement->bindParam(':user_id', $userId);
        $statement->bindParam(':category_id', $preferedCategories[$i]);

        $statement->execute();
    }
}

function userPageCount(&$totalVisits, &$totalVisitsInPercentage,
&$recentVisits, &$recentVisitsInPercentage){
    global $pages;
    $currentTimestamp = time();

    $file = file(LOG_FILE);

    $length = count($pages);
}

```

```

$totalVisits = array_fill(0, $length, 0);
$totalVisitsInPercentage = array_fill(0, $length, 0);
$recentVisits = array_fill(0, $length, 0);
$recentVisitsInPercentage = array_fill(0, $length, 0);

$totalVisitsSum = 0;
$recentVisitsSum = 0;

foreach($file as $row){
    list($pageName,$date,$ipAddress) = explode("\t",$row);

    $pageName = trim($pageName);
    $date = trim($date);
    $ipAddress = trim($ipAddress);

    $index = array_search($pageName, $pages);

    $totalVisits[$index]++;
    $totalVisitsSum++;

    $dateTimestamp = strtotime($date);
    $difference = $currentTimestamp - $dateTimestamp;

    if($difference <= 86400){
        $recentVisits[$index]++;
        $recentVisitsSum++;
    }
}

$onePercentTotalVisits = $totalVisitsSum/100;
$onePercentRecentVisits = $recentVisitsSum/100;
foreach($pages as $index=>$page){
    $totalVisitsInPercentage[$index] =
$totalVisits[$index]/$onePercentTotalVisits;
    $recentVisitsInPercentage[$index] =
$recentVisits[$index]/$onePercentRecentVisits;
}
}

function getLoggedInUsers(){
    $file = file(LOGGED_ID_RECORD);

    $currentTimestamp = time();

    $users = [];

    foreach($file as $index=>$row){
        list($username, $email, $role, $date) = explode("\t",$row);

```

```
$username = trim($username);
$email = trim($email);
$role = trim($role);

$dateTimestamp = strtotime($date);
$difference = $currentTimestamp - $dateTimestamp;

if($difference <= 86400){
    $user[ 'username' ] = $username;
    $user[ 'email' ] = $email;
    $user[ 'role' ] = $role;
    $user[ 'date' ] = $date;

    $users[] = $user;
}
}

return $users;
}
?>
```

Unban-user.php

```
<?php
    session_start();
    ob_start();
    header('Content-type: application/json');

    require_once "../config/connection.php";
    require_once "../global/functions.php";
    require_once "functions.php";

    if($_SERVER['REQUEST_METHOD']=='GET'){
        http_response_code(404);
        redirect("../index.php?page=404");
        exit;
    }

    if(!isset($_SESSION['userId'])){
        http_response_code(401);
        $errorMessageArray[] = 'Access denied!';
        echo json_encode($errorMessageArray);
        exit;
    }

    if($_SESSION['userRoleName'] != 'admin'){
        http_response_code(403);
        $errorMessageArray[] = 'Access denied!';
        echo json_encode($errorMessageArray);
        exit;
    }

    $errorMessageArray = [];

    $userId = $_POST['userId'];

    try{
        unbanUser($userId);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
        exit;
    }

    try{
```

```
$allUsers = selectAllUsers();
}catch(PDOException $ex){
    http_response_code(500);
    $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
    echo json_encode($errorMessageArray);
    exit;
}

$messsage[ ] = $allUsers;
http_response_code(200);
echo json_encode($messsage);
ob_end_flush();
?>
```

Update-user-info.php

```
<?php
    session_start();
    ob_start();
    header('Content-type: application/json');

    require_once "../../config/connection.php";
    require_once "../../global/functions.php";
    require_once 'functions.php';

    if($_SERVER['REQUEST_METHOD']=='GET'){
        http_response_code(404);
        redirect("../index.php?page=404");
        exit;
    }

    $errorMessageArray = [];

    if(!isset($_SESSION['userId'])){
        http_response_code(401);
        $errorMessageArray[] = 'Access denied!';
        echo json_encode($errorMessageArray);
        exit;
    }

    $userId = $_POST['user-id'];
    $previousAddress = $_POST['previous-address'];

    $avatarName = $_FILES['user-avatar']['name'];
    $avatarTmpName = $_FILES['user-avatar']['tmp_name'];
    $avatarSize = $_FILES['user-avatar']['size'];
    $avatarType = pathinfo($_FILES['user-avatar']['name'],
PATHINFO_EXTENSION);

    $firstName = $_POST['first-name-input'];
    $lastName = $_POST['last-name-input'];
    $username = $_POST['username-input'];
    $email = $_POST['email-input'];
    $phone = $_POST['phone-input'];
    $roleId = $_POST['role-input'];
    $roleName = $_POST['role-name'];
    $biographyText = $_POST['biography-input'];

    $addressLine = $_POST['address-line-input'];
```

```

$addressNumber = $_POST['number-input'];
$city = $_POST['city-input'];
$state = $_POST['state-input'];
$zipCode = $_POST['zip-code-input'];
$country = $_POST['country-input'];

$hasAvatar = false;
$hasAddress = false;

$addressId = null;

if(empty($userId)){
    $errorMessageArray[] = 'Invalid input. Please try again.';
}

if(!empty($avatarName)){
    checkAvatar($avatarSize,$avatarType,$errorMessageArray);

    $hasAvatar = true;
} else{
    $avatarObj = doesAvatarAlreadyExists($userId);
}

checkInputElementWithRegex('/^([A-ZŠĐĆŽ][a-zđčćž]{2,})([A-ZŠĐĆŽ][a-zđčćž]{2,})*$/', $firstName, 'Incorrect first name format (ex. Jhon)', $errorMessageArray);
checkInputElementWithRegex('/^([A-ZŠĐĆŽ][a-zđčćž]{2,})*$/', $lastName, 'Incorrect Last name format (ex. Smith)', $errorMessageArray);
checkInputElementWithRegex('/^([a-zA-Z0-9.-_]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,})$', $email, 'Incorrect email format (ex. jhonsmith@gmail.com)', $errorMessageArray);
checkInputElementWithRegex('/^\d{5,15}$/', $phone, 'Incorrect phone format (ex. 0611234567)', $errorMessageArray);

if($roleId != '2' && $roleId != '3'){
    $errorMessageArray[] = "Role is not valid";
} else if($roleId == '3'){
    checkInputElementWithRegex('/(\s.*){4,}/', $biographyText, 'Biography must have at least 5 words.', $errorMessageArray);
}

if(!empty($addressLine) || !empty($addressNumber) || !empty($city) || !empty($state) || !empty($zipCode) || !empty($country)){

```

```

        checkInputElementWithRegex('/^([a-zA-Z\u0100-\u017E]+)$/', $addressLine, 'Incorrect address line', $errorMessageArray);
        checkInputElementWithRegex('/^\d+[a-zA-Z]{2,}$/', $addressNumber, 'Incorrect address number', $errorMessageArray);
        checkInputElementWithRegex('/^([A-Z\u0100-\u017E]{2,}\s?)+$/', $city, 'Incorrect city', $errorMessageArray);
        checkInputElementWithRegex('/^([A-Z\u0100-\u017E]{2,}\s?)+$/', $state, 'Incorrect state', $errorMessageArray);
        checkInputElementWithRegex('/^\d{5,15}$/', $zipCode, 'Incorrect zip code', $errorMessageArray);
        checkInputElementWithRegex('/^([A-Z\u0100-\u017E]{2,}\s?)+$/', $country, 'Incorrect country', $errorMessageArray);

        $hasAddress = true;
    }

    if($errorMessageArray){
        http_response_code(400);
        echo json_encode($errorMessageArray);
        exit;
    }

    if($hasAddress){
        try{
            $addressIdObj =
doesAddressAlreadyExists($addressLine,$addressNumber,$city,$state,$zipC
ode,$country);
        }catch(PDOException $ex){
            http_response_code(500);
            $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
            echo json_encode($errorMessageArray);
            exit;
        }

        if($addressIdObj){
            $addressId = $addressIdObj->id;
        }else{
            try{
                $addressId =
insertNewAddress($addressLine,$addressNumber,$city,$state,$zipCode,$cou
ntry);
            }catch(PDOException $ex){
                http_response_code(500);
                $errorMessageArray[] = 'Server error: ' . $ex-
>getMessage();
                echo json_encode($errorMessageArray);
                exit;
            }
        }
    }
}

```

```

    }

}

if(!empty($previousAddress) && empty($addressLine)){
    $addressId = null;
}

try{
    $addressId =
updateUserInformation($userId,$firstName,$lastName,$username,$email,$ph
one,$addressId);
}catch(PDOException $ex){
    http_response_code(500);
    $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
    echo json_encode($errorMessageArray);
    exit;
}

if($roleId == '3'){
    try{
        updateUserBiography($userId,$biographyText);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
        exit;
    }
}

if($hasAvatar){
    try{
        $avatarObj = doesAvatarAlreadyExists($userId);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
        exit;
    }

    if($avatarObj){
        $uploadDir = '../../assets/images/avatars/';
        $filePath = $uploadDir . $avatarObj->src;

        if (file_exists($filePath)) {
            unlink($filePath);
    }
}

```

```

        $filePath = $uploadDir . $avatarObj->id .
'.' . $avatarType;
        $result = move_uploaded_file($avatarTmpName,$filePath);

        if(!$result){
            http_response_code(500);
            $errorMessageArray[] = 'Server error.';
            echo json_encode($errorMessageArray);
            exit;
        }

        try{
            updateAvatarRowName($avatarObj->id,$avatarType);
        }catch(PDOException $ex){
            http_response_code(500);
            $errorMessageArray[] = 'Server error: ' . $ex-
>getMessage();
            echo json_encode($errorMessageArray);
            exit;
        }

    } else {
        http_response_code(500);
        $errorMessageArray[] = "File doesn't exist";
        echo json_encode($errorMessageArray);
        exit;
    }

}else{
    try{
        $idAvatar = insertNewAvatarRow($userId);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex-
>getMessage();
        echo json_encode($errorMessageArray);
        exit;
    }

    $uploadDir = '../../assets/images/avatars/';
    $filePath = $uploadDir . $idAvatar . '.' . $avatarType;
    $result = move_uploaded_file($avatarTmpName,$filePath);

    if(!$result){
        http_response_code(500);
        $errorMessageArray[] = 'Server error.';
        echo json_encode($errorMessageArray);
        exit;
    }
}

```

```
}

try{
    updateAvatarRowName($idAvatar,$avatarType);
}catch(PDOException $ex){
    http_response_code(500);
$errorMessageArray[] = 'Server error: ' . $ex-
>getMessage();
    echo json_encode($errorMessageArray);
    exit;
}

}

http_response_code(200);
$messageArray[] = true;
echo json_encode($messageArray);
ob_end_flush();
?>
```

Models > writer > functions.php

```
<?php

function selectWriter($writerId){
    global $conn;

    $query =
        "SELECT CONCAT(user.first_name, ' ', user.last_name) AS writer,
        avatar.src, biography.biography_text
        FROM user
        JOIN biography ON user.id = biography.user_id
        JOIN role ON role.id = user.role_id
        LEFT JOIN avatar ON avatar.user_id = user.id
        WHERE user.id = :writer_id";

    $statement = $conn->prepare($query);
    $statement->bindParam(':writer_id', $writerId);
    $statement->execute();

    $result = $statement->fetch();

    return $result;
}

function selectWriterBooks($writerId){
    global $conn;

    $query = "SELECT book.name AS title, image.src AS src, book.id AS
id, CONCAT(user.first_name, ' ', user.last_name) AS writer,
book.user_id AS writer_id, IFNULL(CEIL(AVG(review.stars)),0) AS
review
        FROM book
        JOIN image ON image.id = book.image_id
        JOIN user ON user.id = book.user_id
        LEFT JOIN review ON review.book_id = book.id
        JOIN publisher ON publisher.id = book.publisher_id
        JOIN book_category ON book_category.book_id = book.id
        JOIN category ON category.id = book_category.category_id
        WHERE book.status = 1 AND publisher.status = 1 AND category.status
= 1 AND book.user_id = :writer_id
        GROUP BY book.id ";

    $statement = $conn->prepare($query);
    $statement->bindParam(':writer_id', $writerId);
    $statement->execute();
```

```

$result = $statement->fetchAll();

return $result;
}

function selectWriterCategories($writerId){
    global $conn;

    $query = 'SELECT DISTINCT parent.id, parent.name
    FROM category parent
    JOIN category child ON child.category_id = parent.id
    WHERE parent.status = 1 AND child.id IN (SELECT DISTINCT
book_category.category_id
                                FROM book
                                JOIN book_category ON
                                JOIN category ON
                                WHERE book.user_id
= :writer_id)';
    $statement = $conn->prepare($query);
    $statement->bindParam(':writer_id',$writerId);
    $statement->execute();

    $result = $statement->fetchAll();

    return $result;
}

functionselectAllWriters(){
    global $conn;

    $query = "SELECT CONCAT(user.first_name, ' ',user.last_name) AS
writer, COUNT(book.id) AS book_count, avatar.src , user.id as writer_id
    FROM book
    JOIN user ON user.id = book.user_id
    JOIN avatar ON avatar.user_id = user.id
    WHERE user.status = 1 AND user.is_banned = 0 AND book.status = 1
    GROUP BY book.user_id";

    $statement = $conn->prepare($query);
    $statement->execute();

    $result = $statement->fetchAll();

    return $result;
}

```

```

function displayWriterArticle($writer){
    $display = '<article class="article-writer"><a href="index.php?page=writer&id='.$writer['writer_id'].'">
        </a>
        <h3>'.$writer['writer']. '</h3>
        <p>'.$writer['book_count'].' titles</p>
    </article>';

    return $display;
}

function selectAllWriterBooksForTable($writerId){
    global $conn;

    $query =
        "SELECT image.src AS src, 0 AS categories_count ,image.alt AS alt ,
    publisher.status AS publisher_status , book.name AS name,book.id AS id,book.user_id AS writer_id, book.price AS price,book.status AS status,
    book.release_date AS release_date
        FROM book
        JOIN image ON book.image_id = image.id
        JOIN publisher ON publisher.id = book.publisher_id
        WHERE book.user_id = :writer_id";

    $statement = $conn->prepare($query);
    $statement->bindParam(':writer_id',$writerId);

    $statement->execute();
    $result = $statement->fetchAll();

    $activeBooks = array();
    $deletedBooks = array();

    for($i=0; $i< count($result); $i++){
        $query =
            "SELECT COUNT(book_category.category_id) AS category_count
            FROM book_category JOIN category ON category.id =
book_category.category_id
            WHERE category.status = 1 AND book_category.book_id = :book_id
            GROUP BY book_category.book_id";

        $statement = $conn->prepare($query);
        $statement->bindParam(':book_id',$result[$i]['id']);
        $statement->execute();
        $count = $statement->fetch(PDO::FETCH_OBJ);
    }
}

```

```
if(is_object($count)){
    $result[$i]['categories_count'] = $count->category_count;
}
}

foreach ($result as $book) {
    if ($book['status'] == 1 && $book['publisher_status'] &&
$book['categories_count'] > 0) {
        $activeBooks[] = $book;
    } else {
        $deletedBooks[] = $book;
    }
}

$booksTable = new stdClass();
$booksTable->activeBooks = $activeBooks;
$booksTable->deletedBooks = $deletedBooks;

return $booksTable;
}
?>
```

Footer.php

```
<?php
    require_once "models/footer/functions.php";

    $role = 'unauthorised';
    if(isset($_SESSION['userRoleName'])){
        $role = $_SESSION['userRoleName'];
    }

    $allLinks = allLinks($role);
?>

<footer>
    <div id="footer" class="wrapper">

        <ul id="documents-links">
            <?php foreach($allLinks->documentsLinks as $link):?>
                <li><a href=<?= $link['href'] ?>
                    target=<?= $link['target'] ?><?= $link['name'] ?></a></li>
            <?php endforeach?>
        </ul>

        <ul id="social-media-links">
            <?php foreach($allLinks->socialMediaLinks as $link):?>
                <li><a href=<?= $link['href'] ?>
                    target=_blank><?= $link['name'] ?></a></li>
            <?php endforeach?>
        </ul>

        <ul id="pages-links">
            <?php foreach($allLinks->pageLinks as $link):?>
                <li><a
                    href="index.php?page=<?= $link['href'] ?>"><?= $link['name'] ?></a></li>
            <?php endforeach?>
        </ul>
    </div>
</footer>

<script src="assets/js/main.js"></script>
<script src="https://code.jquery.com/jquery-3.6.4.js"
integrity="sha256-a9jBBRygX1Bh5lt8GZjXDzyOB+bWve9Ei07tROUTj/E="
crossorigin="anonymous"></script>
</body>
```

Head.php

```
<!DOCTYPE html>
<html lang="en">
<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="author"
content="mailto:nikola.samardzic.54.21@ict.edu.rs"/>
    <meta name="keywords" content="Readily, books, online bookstore,
fiction, non-fiction, mystery, romance, travel, children's books, self-
improvement, business and economics, technology and science, history,
cooking"/>
    <meta name="copyright" content="Readily 2022"/>
    <meta name="language" content="english"/>

    <!--Favicon-->
    <link rel="shortcut icon" href="assets/images/favicon.ico"
sizes="16x16" />

    <!--Fonts-->
    <link rel="preconnect" href="https://fonts.googleapis.com" />
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin
/>
    <link
href="https://fonts.googleapis.com/css2?family=Source+Sans+Pro:ital,wght@0,400;0,600;0,700;0,900;1,400;1,600;1,700&family=Source+Serif+Pro:ita
l,wght@0,400;0,600;0,700;0,900;1,400;1,600;1,700&display=swap"
rel="stylesheet" />

    <!--Icons-->
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.2.1/css/all.min.css" integrity="sha512-
MV7K8+y+gLIBoVD59lQIYicR65iaqukzvf/nwasF0nqhPay5w/91JmVM2hMDcnK10nMGCdV
K+iQrJ7lzPJQd1w==" crossorigin="anonymous" referrerPolicy="no-referrer"
/>

    <link rel="stylesheet" href="assets/css/style.css" />
```

Header.php

```
<?php
    require_once "models/header/functions.php";
    require_once "models/global/functions.php";

    $role = 'unauthorised';
    if(isset($_SESSION['userRoleName'])){
        $role = $_SESSION['userRoleName'];
        $avatar_src = isset($_SESSION['userAvatarSrc']) ?
$_SESSION['userAvatarSrc'] : 'default-avatar.jpg';
    }

    $links = headerLinks($role);
    global $current_page;

    $page = isset($_GET['page']) ? $_GET['page'] : "home";

    recordUserVisit();
?>

<header>
    <div id="header" class="wrapper">
        <a href="index.php?page=home">
            
        </a>
        <ul id="header-navigation">
            <?php foreach($links as $link):?>
                <li><a href="index.php?page=<?=$link['href'] ?>">
                    <?php if($page == $link['href'])echo('id="active-link"'); ?><?=$link['name']?></a></li>
                <?php endforeach?>
            </ul>
            <div id="icons-container">

                <?php if(isset($_SESSION['userId'])){>
                    echo '
                    <div>
                        <a href="index.php?page=user-profile">
                            
                        </a>
                    </div>
                '>;
            <?php } ?>
        </div>
    </div>
</header>
```

```
        <i class="fa-solid fa-moon" id="dark-mode-icon"
style="display: none;"></i>
        <i class="fa-solid fa-sun" id="light-mode-icon"
style="display: block;"></i>
            <i class="fa-solid fa-bars" id="menu-icon-open"></i>
        </div>
    </div>
</header>

<div class="phone-nav" id="phone-nav-id">
    <ul id="phone-nav-ul">
        <?php foreach($links as $link):?>
            <li><a href=<?= $link['href'] ?>">
                <?php if($page == $link['href'])echo('id="active-
link"'); ?><?= $link['name']?></a></li>
            <?php endforeach?>
        </ul>
        <i id="menu-icon-close" class="fa-solid fa-xmark"></i>
    </div>
```

404.php

```
<title>404</title>
</head>
<body>

<section id="four-o-four">
<h1>404</h1>
</section>
```

Admin-categories.php

```
<?php
    require_once "models/global/functions.php";
    require_once "models/category/functions.php";

    if(!isset($_SESSION["userRoleName"]) || 
($_SESSION["userRoleName"] != "admin")){
        http_response_code(404);
        redirect('index.php?page=404');
        exit;
    }

    try{
        $categoriesTable = selectAllCategories();
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
    }

    try{
        $allParentCategories = selectAllActiveCategories(null);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
    }
?>

<title>Admin Categories | Readily</title>
</head>
<body>

<section id="admin-categories-section" class="wrapper">

    <form class="insert-update-row-form" id="categories-form"
style="display: none;">

        <label style="display: none;" id="select-category-label"
for="select-category">Parent Category</label>
        <select style="display:none" name="select-category" id="select-
category">
            <?php foreach ($allParentCategories as $category): ?>
```

```

        <option
value=<?= $category[ 'id' ] ?>><?= $category[ 'name' ]?></option>

        <?php endforeach; ?>
    </select>

    <label for="category-name" >Name</label>
    <input type="text" name="category-type" id="category-type"
value="subcategory" hidden>
        <input type="text" name="category-id" id="category-id"
value="0" hidden>
        <input type="text" id="category-name" name="category-name">

        <p id="category-name-error" class="error-
message" style="display: none;">Incorrect format</p>
        <div class="button-option-container">
            <input type="button" class="danger-option" id="cancel-
button" value="Cancel">
            <input type="button" class="safe-option" id="save-button"
value="Save">
        </div>
    </form>

<div class="server-messages" style="display:none"></div>

<h2>Active Categories</h2>

<div class="table-container">
    <table id="table-active-categories">
        <thead>
            <tr>
                <th>Name</th>
                <th>Parent Category</th>
                <th>Book Count</th>
                <th>Created At</th>
                <th>Update</th>
                <th>Delete</th>
            </tr>
        </thead>
        <tbody>
            <?php foreach ($categoriesTable->activeCategories as
$category): ?>
            <tr>
                <td><?php echo $category[ 'name' ]; ?></td>
                <td><?= empty($category[ 'parent' ]) ?>
"/" :$category[ 'parent' ] ?></td>
                <td><?= $category[ 'book_count' ] ?></td>
                <td><?php echo $category[ 'created_at' ]; ?></td>
            </tr>
        
```

```

        <td><a data-category-parent-
id=<?=$category['category_id']?>" data-category-
id=<?=$category['id']?>" data-category-type=<?=
empty($category['parent']) ? 'category' : 'subcategory' ?>" data-
category-name=<?=$category['name']?>" class="safe-option change-links"
href="#">Change</a></td>
        <td>
            <form method="post" name="delivery-form" >
                <input type="text"
value=<?=$category['id'] ?>" hidden name="category-id">
                <input class="danger-option delete-
category" type="button" value="Delete">
            </form>
        </td>
    </tr>
    <?php endforeach; ?>
</tbody>
</table>
</div>

<div class="add-button-container add-buttons-container" >
    <input class="safe-option" id="add-subcategory-button"
type="button" value="Add Subcategory">
    <input class="safe-option" id="add-category-button"
type="button" value="Add Category">
</div>

<h2>Deleted Publishers</h2>

<div class="table-container">
    <table id="table-deleted-categories">
        <thead>
            <tr>
                <th>Name</th>
                <th>Parent Category</th>
                <th>Book Count</th>
                <th>Deleted At</th>
                <th>Update</th>
                <th>Activate</th>
            </tr>
        </thead>
        <tbody>
            <?php foreach ($categoriesTable->deletedCategories as
$category): ?>
            <tr>
                <td><?php echo $category['name']; ?></td>

```

```

        <td><?= empty($category['parent']) ?>
"/" :$category['parent'] ?></td>
        <td>3</td>
        <td><?php echo $category['deleted_at']; ?></td>
        <td><a data-category-parent-
id=<?=$category['category_id']?>" data-category-
id=<?=$category['id']?>" data-category-name=<?=$category['name']?>"-
data-category-type=<?= empty($category['parent']) ? 'category' :
'subcategory' ?>" class="safe-option change-links"
href="#">Change</a></td>
        <td>
            <form method="post" name="delivery-form" >
                <input type="text"
value=<?=$category['id'] ?>" hidden name="category-id">
                <input class="safe-option activate-
category" type="button" value="Activate">
            </form>
        </td>
    </tr>
<?php endforeach; ?>
</tbody>
</table>
</div>
</section>

```

Admin-delivery-options.php

```
<?php
    require_once "models/global/functions.php";
    require_once "models/delivery/functions.php";

    if(!isset($_SESSION["userRoleName"]) || 
($_SESSION["userRoleName"] != "admin")){
        http_response_code(404);
        redirect('index.php?page=404');
        exit;
    }

    try{
        $deliveryTypeTable = selectAllDeliveryTypes();
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
    }
?>

<title>Admin Delivery Types | Readily</title>
</head>
<body>

<section id="admin-publishers-section" class="wrapper">

    <form class="insert-update-row-form" id="delivery-type-form"
style="display: none;">
        <label for="delivery-type-name" >Name</label>
        <input type="text" name="delivery-type-id" id="delivery-type-
id" value="0" hidden>
        <input type="text" id="delivery-type-name" name="delivery-type-
name">
        <p id="delivery-type-name-error" class="error-
message" style="display: none;">Incorrect format</p>
        <div class="button-option-container">
            <input type="button" class="danger-option" id="cancel-
button" value="Cancel">
            <input type="button" class="safe-option" id="save-button"
value="Save">
        </div>
    </form>

```

```

<div class="server-messages" style="display:none"></div>

<h2>Active Delivery Types</h2>

<div class="table-container">
    <table id="table-active-delivery-types">
        <thead>
            <tr>
                <th>Name</th>
                <th>Orders Count</th>
                <th>Created At</th>
                <th>Update</th>
                <th>Delete</th>
            </tr>
        </thead>
        <tbody>
            <?php foreach ($deliveryTypeTable->activeDeliveryTypes
as $deliveryType): ?>
            <tr>
                <td><?php echo $deliveryType['name']; ?></td>
                <td><?php echo
$deliveryType['order_count']; ?></td>
                <td><?php echo
$deliveryType['created_at']; ?></td>
                <td><a data-delivery-type-
id="<?=$deliveryType['id']?>" data-delivery-type-
name="<?=$deliveryType['name']?>" class="safe-option change-links"
href="#">Change</a></td>
                <td>
                    <form method="post" name="delivery-form" >
                        <input type="text"
value="<?=$deliveryType['id'] ?>" hidden name="delivery-type-id">
                        <input class="danger-option delete-
delivery-type" type="button" value="Delete">
                    </form>
                </td>
            </tr>
            <?php endforeach; ?>
        </tbody>
    </table>
</div>

<div class="add-button-container add-buttons-container" >
    <input class="safe-option" id="add-delivery-type-button"
type="button" value="Add Delivery Type">
</div>

<h2>Deleted Delivery Types</h2>

```

```

<div class="table-container">
    <table id="table-deleted-delivery-types">
        <thead>
            <tr>
                <th>Name</th>
                <th>Orders Count</th>
                <th>Deleted At</th>
                <th>Update</th>
                <th>Activate</th>
            </tr>
        </thead>
        <tbody>
            <?php foreach ($deliveryTypeTable->deletedDeliveryTypes
as $deliveryType): ?>
            <tr>
                <td><?php echo $deliveryType['name']; ?></td>
                <td><?php echo
$deliveryType['order_count']; ?></td>
                <td><?php echo
$deliveryType['deleted_at']; ?></td>
                <td><a data-delivery-type-
id="<?=$deliveryType['id']?>" data-delivery-type-
name="<?=$deliveryType['name']?>" class="safe-option change-links"
href="#">Change</a></td>
                <td>
                    <form method="post" name="delivery-form" >
                        <input type="text"
value="<?=$deliveryType['id'] ?>" hidden name="delivery-type-id">
                        <input class="safe-option activate-
delivery-type" type="button" value="Activate">
                    </form>
                </td>
            </tr>
        <?php endforeach; ?>
    </tbody>
</table>
</div>
</section>

```

Admin-logged-id-users.php

```
<?php
    require_once "models/global/functions.php";
    require_once "models/user/functions.php";

    if(!isset($_SESSION["userRoleName"]) || 
($_SESSION["userRoleName"] != "admin")){
        http_response_code(404);
        redirect('index.php?page=404');
        exit;
    }

    $loggedInUsers = getLoggedInUsers();
?>

<title>Admin Logged-in Users | Readily</title>
</head>
<body>

<section id="admin-users-section" class="wrapper">
    <h2>Logged-in Users</h2>

    <div class="table-container">
        <table id="table-active-users">
            <thead>
                <tr>
                    <th>Username</th>
                    <th>Email</th>
                    <th>Role</th>
                    <th>Date</th>
                </tr>
            </thead>
            <tbody>
                <?php foreach ($loggedInUsers as $user): ?>
                <tr>
                    <td><?=$user['username'] ?></td>
                    <td><?=$user['email'] ?></td>
                    <td><?=$user['role'] ?></td>
                    <td><?=$user['date'] ?></td>
                </tr>
                <?php endforeach; ?>
            </tbody>
        </table>
    </div>
```

```
</section>
```

Admin-messages.php

```
<?php
    require_once "models/global/functions.php";
    require_once "models/message/functions.php";

    if(!isset($_SESSION["userRoleName"]) || 
($_SESSION["userRoleName"] != "admin")){
        http_response_code(404);
        redirect('index.php?page=404');
        exit;
    }

    try{
        $userMessages = getUserMessages();
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
    }
?>

<title>Admin Messages | Readily</title>
</head>
<body>

<section id="admin-messages-section" class="wrapper">

    <h2>User Messages</h2>

    <div class="table-container">
        <table id="table-active-publishers">
            <thead>
                <tr>
                    <th>Username</th>
                    <th>Email</th>
                    <th>Created At</th>
                    <th>Subject</th>
                    <th>Text</th>
                </tr>
            </thead>
            <tbody>
                <?php foreach ($userMessages as $message): ?>
                <tr>
                    <td><?=$message['username']; ?></td>
```

```
<td><?=$message['email'] ?></td>
<td><?=$message['date']; ?></td>
<td><?=$message['subject'] ?></td>
<td class="td-
text"><?=$message['text'] ?></td>
</tr>
<?php endforeach; ?>
</tbody>
</table>
</div>

</section>
```

Admin-orders.php

```
<?php
    require_once "models/global/functions.php";
    require_once "models/order/functions.php";

    if(!isset($_SESSION["userRoleName"]) || 
($_SESSION["userRoleName"] != "admin")){
        http_response_code(404);
        redirect('index.php?page=404');
        exit;
    }

    try{
        $userOrders = getUserOrders();
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
    }
?>

<title>Admin Orders | Readily</title>
</head>
<body>

<section id="admin-messages-section" class="wrapper">

    <h2>User Orders</h2>

    <div class="table-container">
        <table id="table-active-publishers">
            <thead>
                <tr>
                    <th>Username</th>
                    <th>Email</th>
                    <th>Finished At</th>
                    <th>Price</th>
                    <th>Status</th>
                </tr>
            </thead>
            <tbody>
                <?php foreach ($userOrders as $order): ?>
                <tr>
                    <td><?=$order['username']; ?></td>
```

```
<td><?=$order['email'] ?></td>
<td><?=$order['finished_at']; ?></td>
<td>$ <?=$order['total_price'] ?>.00</td>
<td><?=$order['name'] ?></td>
</tr>
<?php endforeach; ?>
</tbody>
</table>
</div>

</section>
```

Admin-publishers.php

```
<?php
    require_once "models/global/functions.php";
    require_once "models/publisher/functions.php";

    if(!isset($_SESSION["userRoleName"]) || 
($_SESSION["userRoleName"] != "admin")){
        http_response_code(404);
        redirect('index.php?page=404');
        exit;
    }

    try{
        $publisherTable = selectAllPublishers();
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
    }
?>

<title>Admin Publishers | Readily</title>
</head>
<body>

<section id="admin-publishers-section" class="wrapper">

    <form class="insert-update-row-form" id="publisher-form"
style="display: none;">
        <label for="publisher-name" >Name</label>
        <input type="text" name="publisher-id" id="publisher-id"
value="0" hidden>
        <input type="text" id="publisher-name" name="publisher-name">
        <p id="publisher-name-error" class="error-
message" style="display: none;">Incorrect format</p>
        <div class="button-option-container">
            <input type="button" class="danger-option" id="cancel-
button" value="Cancel">
            <input type="button" class="safe-option" id="save-button"
value="Save">
        </div>
    </form>

    <div class="server-messages" style="display:none"></div>
```

```

<h2>Active Publishers</h2>

<div class="table-container">
    <table id="table-active-publishers">
        <thead>
            <tr>
                <th>Name</th>
                <th>Book Count</th>
                <th>Created At</th>
                <th>Update</th>
                <th>Delete</th>
            </tr>
        </thead>
        <tbody>
            <?php foreach ($publisherTable->activePublishers as $publisher): ?>
            <tr>
                <td><?php echo $publisher['name']; ?></td>
                <td><?=$publisher['book_count'] ?></td>
                <td><?php echo
$publisher['created_at']; ?></td>
                    <td><a data-publisher-
id="<?=$publisher['id']?>" data-publisher-
name="<?=$publisher['name']?>" class="safe-option change-links"
href="#">Change</a></td>
                    <td>
                        <form method="post" name="delivery-form" >
                            <input type="text"
value="<?=$publisher['id'] ?>" hidden name="publisher-id">
                            <input class="danger-option delete-
publisher" type="button" value="Delete">
                        </form>
                    </td>
            </tr>
            <?php endforeach; ?>
        </tbody>
    </table>
</div>

<div class="add-button-container add-buttons-container" >
    <input class="safe-option" id="add-publisher-button"
type="button" value="Add Publisher">
</div>

<h2>Deleted Publishers</h2>

<div class="table-container">
    <table id="table-deleted-publishers">

```

```

<thead>
    <tr>
        <th>Name</th>
        <th>Book Count</th>
        <th>Deleted At</th>
        <th>Update</th>
        <th>Activate</th>
    </tr>
</thead>
<tbody>
    <?php foreach ($publisherTable->deletedPublishers as $publisher): ?>
        <tr>
            <td><?php echo $publisher['name']; ?></td>
            <td><?=$publisher['book_count'] ?></td>
            <td><?php echo
$publisher['deleted_at']; ?></td>
                <td><a data-publisher-
id=<?=$publisher['id']?>" data-publisher-
name=<?=$publisher['name']?>" class="safe-option change-links"
href="#">Change</a></td>
            <td>
                <form method="post" name="delivery-form" >
                    <input type="text"
value=<?=$publisher['id'] ?>" hidden name="publisher-id">
                    <input class="safe-option activate-
publisher" type="button" value="Activate">
                </form>
            </td>
        </tr>
    <?php endforeach; ?>
</tbody>
</table>
</div>
</section>

```

Admin-survey.php

```
<?php
    require_once "models/global/functions.php";
    require_once "models/survey/functions.php";

    if(!isset($_SESSION["userRoleName"])) ||
($_SESSION["userRoleName"] != "admin")){
        http_response_code(404);
        redirect('index.php?page=404');
        exit;
    }

    try{
        $surveyTable = selectAllSurveys();
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
    }
?>

<title>Admin Survey | Readily</title>
</head>
<body>

<section id="admin-survey-section" class="wrapper">

    <h2>Active Surveys</h2>

    <div class="table-container">
        <table id="table-active-surveys">
            <thead>
                <tr>
                    <th>Name</th>
                    <th>Created At</th>
                    <th class="survey-table-options">Option 1</th>
                    <th class="survey-table-options">Option 2</th>
                    <th class="survey-table-options">Option 3</th>
                    <th class="survey-table-options">Option 4</th>
                    <th class="survey-table-options">Option 5</th>
                    <th>Delete</th>
                </tr>
            </thead>
            <tbody>
```

```

<?php foreach ($surveyTable->activeSurveys as
$survey): ?>
    <tr>
        <td><?php echo $survey['name']; ?></td>
        <td><?= $survey['created_at'] ?></td>

        <?php foreach($survey['options'] as $optionName
=> $optionValue): ?>
            <td class="survey-table-options">
                <p><?=$optionName ?></p>
                <p><?=$optionValue ?></p>
            </td>
            <?php endforeach ?>

            <td>
                <form method="post" name="delivery-form" >
                    <input type="text"
value="<?=$survey['id'] ?>" hidden name="survey-id">
                    <input class="danger-option delete-
survey" type="button" value="Delete">
                </form>
            </td>
        </tr>
    <?php endforeach; ?>
</tbody>
</table>
</div>

<h2>Finished Surveys</h2>

<div class="table-container">
    <table id="table-deleted-surveys">
        <thead>
            <tr>
                <th>Name</th>
                <th>Finished At</th>
                <th class="survey-table-options">Option 1</th>
                <th class="survey-table-options">Option 2</th>
                <th class="survey-table-options">Option 3</th>
                <th class="survey-table-options">Option 4</th>
                <th class="survey-table-options">Option 5</th>
                <th>Activate</th>
            </tr>
        </thead>
        <tbody>
            <?php foreach ($surveyTable->deletedSurveys as
$survey): ?>
                <tr>

```

```
<td><?php echo $survey['name']; ?></td>
<td><?= $survey['finished_at'] ?></td>

<?php foreach($survey['options'] as $optionName
=> $optionValue): ?>
    <td class="survey-table-options">
        <p><?= $optionName ?></p>
        <p><?= $optionValue ?></p>
    </td>
    <?php endforeach ?>

    <td>
        <form method="post" name="delivery-form" >
            <input type="text"
value="<?= $survey['id'] ?>" hidden name="survey-id">
            <input class="safe-option activate-
survey" type="button" value="Activate">
        </form>
    </td>
</tr>
<?php endforeach; ?>
</tbody>
</table>
</div>
</section>
```

Admin-users.php

```
<?php
    require_once "models/global/functions.php";
    require_once "models/user/functions.php";

    if(!isset($_SESSION["userRoleName"]) || 
($_SESSION["userRoleName"] != "admin")){
        http_response_code(404);
        redirect('index.php?page=404');
        exit;
    }

    try{
        $userTable = selectAllUsers();
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
    }
?>

<title>Admin Users | Readily</title>
</head>
<body>

<div class="wrapper server-messages " id="admin-users-server-messages">
</div>

<section id="admin-users-section" class="wrapper">
    <h2>Active Users</h2>

    <div class="table-container">
        <table id="table-active-users">
            <thead>
                <tr>
                    <th>Username</th>
                    <th>Email</th>
                    <th>Role</th>
                    <th>Created At</th>
                    <th>Update</th>
                    <th>Ban</th>
                    <th>Delete</th>
                </tr>
            </thead>
```

```

<tbody>
    <?php foreach ($userTable->activeUsers as $user): ?>
        <tr>
            <td><?php echo $user['username']; ?></td>
            <td><?php echo $user['email']; ?></td>
            <td><?php echo $user['name']; ?></td>
            <td><?php echo $user['created_at']; ?></td>
            <td><a class="safe-option"
href="index.php?page=user-profile&selected-
user=<?=$user['id']?>">Change</a></td>
            <td>
                <form method="post" name="delivery-form" >
                    <input type="text"
value="<?=$user['id'] ?>" hidden name="userId">
                    <input class="danger-option ban-user"
type="button" value="Ban">
                </form>
            </td>
            <td>
                <form method="post" name="delivery-form" >
                    <input type="text"
value="<?=$user['id'] ?>" hidden name="userId">
                    <input class="danger-option delete-
user" type="button" value="Delete">
                </form>
            </td>
        </tr>
    <?php endforeach; ?>
</tbody>
</table>
</div>

<h2>Deleted Users</h2>
<div class="table-container">
    <table id="table-deleted-users">
        <thead>
            <tr>
                <th>Username</th>
                <th>Email</th>
                <th>Role</th>
                <th>Created At</th>
                <th>Deleted At</th>
                <th>Update</th>
                <th>Activate</th>
            </tr>
        </thead>
        <tbody>
            <?php foreach ($userTable->deletedUsers as $user): ?>

```

```

<tr>
    <td><?php echo $user['username']; ?></td>
    <td><?php echo $user['email']; ?></td>
    <td><?php echo $user['name']; ?></td>
    <td><?php echo $user['created_at']; ?></td>
    <td><?php echo $user['deleted_at']; ?></td>
    <td><a class="safe-option"
href="index.php?page=user-profile&selected-
user=<?= $user['id'] ?>">Change</a></td>

    <td>
        <form method="post" name="delivery-form" >
            <input type="text"
value="<?= $user['id'] ?>" hidden name="userId">
            <input class="safe-option activate-
user" type="button" value="Activate">
        </form>
    </td>
</tr>
<?php endforeach; ?>
</tbody>
</table>
</div>

<h2>Banned Users</h2>
<div class="table-container">
    <table id="table-banned-users">
        <thead>
            <tr>
                <th>Username</th>
                <th>Email</th>
                <th>Role</th>
                <th>Created At</th>
                <th>Banned At</th>
                <th>Update</th>
                <th>Unban</th>
            </tr>
        </thead>
        <tbody>
            <?php foreach ($userTable->bannedUsers as $user): ?>
            <tr>
                <td><?php echo $user['username']; ?></td>
                <td><?php echo $user['email']; ?></td>
                <td><?php echo $user['name']; ?></td>
                <td><?php echo $user['created_at']; ?></td>
                <td><?php echo $user['banned_at']; ?></td>

```

```
<td><a class="safe-option"
href="index.php?page=user-profile&selected-
user=<?=$user['id']?>">Change</a></td>
<td>
    <form method="post" name="delivery-form" >
        <input type="text"
value="<?=$user['id'] ?>" hidden name="userId">
        <input class="safe-option unban-user"
type="button" value="Unban">
    </form>
</td>

</tr>
<?php endforeach; ?>
</tbody>
</table>
</div>
</section>
```

Admin-visitors.php

```
<?php
    require_once "models/global/functions.php";
    require_once "models/user/functions.php";

    if(!isset($_SESSION["userRoleName"]) || 
($_SESSION["userRoleName"] != "admin")){
        http_response_code(404);
        redirect('index.php?page=404');
        exit;
    }

    $totalVisits;
    $totalVisitsInPercentage;
    $recentVisits;
    $recentVisitsInPercentage;

    userPageCount($totalVisits, $totalVisitsInPercentage, $recentVisits,
$recentVisitsInPercentage);

    global $pages;
?>

<title>Admin Visitors | Readily</title>
</head>
<body>

<div class="wrapper server-messages " id="admin-visitors-server-
messages">
</div>

<section id="admin-users-section" class="wrapper">
    <h2>Visits</h2>

    <div class="table-container">
        <table id="table-active-users">
            <thead>
                <tr>
                    <th>Page</th>
                    <th>Total visits</th>
                    <th>Total visits in %</th>
                    <th>Recent visits</th>
                    <th>Recent visits in %</th>
                </tr>
            </thead>
            <tbody>
                <tr>
                    <td>1</td>
                    <td>100</td>
                    <td>100%</td>
                    <td>100</td>
                    <td>100%</td>
                </tr>
            </tbody>
        </table>
    </div>
</section>
```

```
</thead>
<tbody>
    <?php foreach ($pages as $index=>$page): ?>
        <tr>
            <td><?=$pages[$index] ?></td>
            <td><?=$totalVisits[$index] ?></td>
            <td><?=round((float)$totalVisitsInPercentage[$index],2) ?>%</td>
            <td><?=$recentVisits[$index] ?></td>
            <td><?=round((float)$recentVisitsInPercentage[$index],2) ?>%</td>
        </tr>
    <?php endforeach; ?>
</tbody>
</table>
</div>

</section>
```

Admin.php

```
<?php
    require_once "models/global/functions.php";

    if(!isset($_SESSION["userRoleName"]) || 
($_SESSION["userRoleName"] != "admin")){
        http_response_code(404);
        redirect('index.php?page=404');
        exit;
    }
?>

<title>Admin | Readily</title>
</head>
<body>

<section id="admin-section" class="wrapper">

    <h2 class="admin-option-title">Tables</h2>
    <div class="admin-option-container">
        <div class="admin-option">
            <a href="index.php?page=admin-users"></a>
            <div class="option-name-container">
                <p>Users</p>
            </div>
            <div class="option-icon-container">
                
            </div>
        </div>
        <div class="admin-option">
            <a href="index.php?page=admin-categories"></a>
            <div class="option-name-container">
                <p>Categories</p>
            </div>
            <div class="option-icon-container">
                
            </div>
        </div>
        <div class="admin-option">
            <a href="index.php?page=admin-publishers"></a>
            <div class="option-name-container">
                <p>Publishers</p>
```

```

        </div>
        <div class="option-icon-container">
            
        </div>
    </div>
    <div class="admin-option">
        <a href="index.php?page=admin-delivery-options"></a>
        <div class="option-name-container">
            <p>Delivery Options</p>
        </div>
        <div class="option-icon-container">
            
        </div>
    </div>

    <h2 class="admin-option-title">Surveys</h2>
    <div class="admin-option-container">

        <div class="admin-option">
            <a href="index.php?page=admin-survey"></a>
            <div class="option-name-container">
                <p>Survey Result</p>
            </div>
            <div class="option-icon-container">
                
            </div>
        </div>
    </div>

    <h2 class="admin-option-title">Messages</h2>
    <div class="admin-option-container">

        <div class="admin-option">
            <a href="index.php?page=admin-messages"></a>
            <div class="option-name-container">
                <p>User Messages</p>
            </div>
            <div class="option-icon-container">
                
            </div>
        </div>
    </div>
</div>

```

```
<h2 class="admin-option-title">Statistics</h2>
<div class="admin-option-container">
<div class="admin-option">
    <a href="index.php?page=admin-orders"></a>
    <div class="option-name-container">
        <p>Orders</p>
    </div>
    <div class="option-icon-container">
        
    </div>
</div>

<div class="admin-option">
    <a href="index.php?page=admin-visitors"></a>
    <div class="option-name-container">
        <p>Visitors</p>
    </div>
    <div class="option-icon-container">
        
    </div>
</div>

<div class="admin-option">
    <a href="index.php?page=admin-logged-id-users"></a>
    <div class="option-name-container">
        <p>Logged-in Users</p>
    </div>
    <div class="option-icon-container">
        
    </div>
</div>

</div>
</section>
```

Author.php

```
<title>Author | Readily</title>
</head>
<body>

<section id="author-section">

    <h1>Author</h1>

    <div id="img-container" class="set-brightness">
        
    </div>

    <h2>First name</h2>
    <p>Nikola</p>

    <h2>Last Name</h2>
    <p>Samardžić</p>

    <h2>Index</h2>
    <p>54/21</p>

</section>
```

Book.php

```
<?php
    require_once "models/global/functions.php";
    require_once "models/book/functions.php";
    require_once "models/category/functions.php";
    require_once "models/writer/functions.php";

    if(!isset($_GET['id'])){
        http_response_code(400);
        redirect("index.php?page=home");
        exit;
    }

    $bookId = $_GET['id'];

    try{
        $bookInformation = selectBook($bookId);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
    }

    try{
        $relatedCategories = selectRelatedCategoriesToABook($bookId);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
    }

    try{
        $relatedBooks =
selectRelatedBooksToACategories($relatedCategories);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
    }

    shuffle($relatedBooks);

    try{
        $relatedCategories = getCategories($relatedCategories);
```

```

}catch(PDOException $ex){
    http_response_code(500);
    $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
    echo json_encode($errorMessageArray);
}

try{
    $allWritersBooks =
selectWriterBooks($bookInformation['writer_id']);
}catch(PDOException $ex){
    http_response_code(500);
    $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
    echo json_encode($errorMessageArray);
}

shuffle($allWritersBooks);

try{
    $allTheComments = getBookComments($bookId);
}catch(PDOException $ex){
    http_response_code(500);
    $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
    echo json_encode($errorMessageArray);
}

if(isset($_SESSION['userId'])){
    $userId = $_SESSION['userId'];

    try{
        $review = getSingleReview($userId,$bookId);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
    }
}
}

?>

<title>Writer | Readily</title>
</head>
<body>

<section id="book-info" class="wrapper">
    <h1><?=$bookInformation['title'] ?></h1>
    <p id="writer-link">By <a
href="index.php?page=writer&id=<?=$bookInformation['writer_id']?>"><?=
$bookInformation['writer']?></a></p>

```

```

<div id="stars">
    <?php
    for($i=0;$i<5;$i++){
        if($bookInformation['review'] > $i){
            echo '<i class="fa-solid fa-star"></i>';
        }else{
            echo '<i class="fa-regular fa-star"></i>';
        }
    }
    ?>
    <p><?php
        if($bookInformation['review_count']){
            echo $bookInformation['review']."/5 (<span class='text-
bold'>".$bookInformation['review_count']."' ratings</span>)";}
        else{
            echo "no ratings";
        }
    ?></p>
</div>

<div id="text-container-about">
    <h2>About this book</h2>
    <p style="white-space: pre-
wrap;"><?=$bookInformation['description'] ?></p>
</div>

<section id="related-categories-container" class="section-
articles">
    <h2 class="section-heading">Related Categories</h2>
    <div class="article-container article-category-container"
id="book-related-categories-section">
        <?php foreach($relatedCategories as $category): ?>

            <?=displayCategoryArticle($category) ?>

        <?php endforeach ?>
    </div>
</section>

<div class="section-articles" id="div-info-section">
    <h2 class="section-heading">Info</h2>
    <div class="article-container article-info-container" id="div-
info-container">
        <article>
            <p class="heading-info">Pages</p>
            <p class="text-
bold"><?=$bookInformation['page_count']?></p>
        </article>

```

```

        <article>
            <p class="heading-info">Publisher</p>
            <p class="text-
bold"><?=$bookInformation['publisher_name']?></p>
        </article>
        <article>
            <p class="heading-info">Release Date</p>
            <p class="text-bold"><?=$bookInformation['release_date']?></p>
        </article>
    </div>
</div>

<div id="container-div-image-and-price">
    <div id="container-div-img" class="bg-article-color-
<?=$bookInformation['id']?>">
        ">
    </div>
    <div id="price-container">
        <p>$<?=$bookInformation['price']?>.00</p>
        <i id="book-id-<?=$bookInformation['id'] ?>" class="fa-
solid fa-cart-shopping"></i>
    </div>
</div>

</section>

<section id="related-books" class="wrapper section-articles">
    <h2 class="section-heading">Related to
<?=$bookInformation['title']?></h2>

    <div class="article-container article-book-container" id="related-
books-articles-container">
        <?php foreach($relatedBooks as $book): ?>
            <?=displayBookArticle($book) ?>
        <?php endforeach ?>
    </div>
</section>

<section id="authors-books" class="wrapper section-articles">
    <h2 class="section-heading">Read more from
<?=$bookInformation['writer'] ?></h2>

```

```

<div class="article-container article-book-container" id="autors-
books-articles-container">

    <?php foreach($allWritersBooks as $book): ?>

        <?=displayBookArticle($book) ?>

    <?php endforeach ?>
</div>
</section>

<section id="review-section" class="wrapper">
    <div id="review-container">
        <h2>Reviews for <?=$bookInformation['title'] ?></h2>
        <div id="review-stars-container">

            <?php
            for($i=0;$i<5;$i++){
                if($bookInformation['review'] > $i){
                    echo '<i class="fa-solid fa-star"></i>';
                }else{
                    echo '<i class="fa-regular fa-star"></i>';
                }
            }
            if($bookInformation['review']) echo
            "<p>".$bookInformation['review']."/5</p>";
            ?>
        </div>
        <div id="review-info">
            <p><?=$bookInformation['review_count'] ?> ratings</p>
            <p><?=$bookInformation['comment_count'] ?> reviews</p>
        </div>
    </div>

    <?php if(isset($_SESSION['userId'])):>

        <form id="star-rating-form" class="wrapper" >

            <h2>Rate this Book</h2>

            <div class="star-raiting-container">

                <?php
                for($i=0;$i<5;$i++){
                    echo "
                    <div class='star-icon-container'>
                        <i class='";
                    echo $review['stars']? 'fa-solid' : 'fa-regular';

```

```

        echo " fa-star book-rating-star'></i>
            <input type='radio' value='$i' hidden name='book-stars'
/>
            <input type='text' value='".$bookInformation['id']."' hidden name='book-id' />
        </div>

    ";

}

?>

</div>

<div class="server-messages" style="display:none">
</div>

</form>

<form class="sign-up-container" id="form-comment"
enctype="multipart/form-data">
    <h2>Write a Review</h2>

        <div class="info-container-grid personal-info-container-grid"
id="comment-container">
            <label for="biography-input-js" id="biography-title-js"
class="input-title biography-title" >Comment</label>
            <textarea name="comment-input" id="biography-input-js"
class="biography-input" ></textarea>
            <p id="biography-error" class="error-message" style="display:
none;">There must be at least 5 words.</p>
        </div>

        <p class="p-title">Images</p>
        <div id="comment-images-container">
            <input type="file" class="comment-image" name="comment-
image[]">

        </div>
            <p id="comment-images-error" class="error-message "
style="display: none;">Supported image formats include JPG, JPEG, and
PNG, with a total file size limit of 2MB for all images combined.</p>

        <input type="text" hidden name="book-id"
value="<?=$bookInformation['id']?>">

```

```

        <div class="server-messages comment-server-
error" style="display:none"></div>

        <div class="button-option-container">
            <input type="button" class="safe-option" id="comment-
button" value="Submit">
        </div>
    </form>
<?php endif?>

<div id="comments-container-div">

    <?php foreach($allTheComments as $comment):?>
        <article class="article-comment">
            <div class="comment-user-info-container">
                <div class="avatar-and-name-container">
                    
                    <p class="comment-
username"><?=$comment['username'] ?></p>
                </div>
                <?php if($comment['stars']):?>

                    <div class="comments-stars">
                        <?php
                            for($i=0;$i<5;$i++){
                                if($comment['stars']>$i){
                                    echo '<i class="fa-solid fa-star"></i>';
                                }else{
                                    echo '<i class="fa-regular fa-
star"></i>';
                                }
                            }
                        ?>
                        <p><?=$comment['stars']?>/5</p>
                    </div>
                    <?php endif?>
                </div>

                <div class="text-and-pictures-container">
                    <div class="comment-text">
                        <p class="comment-text-
row"><?=$comment['comment_text']?></p>
                    </div>
                    <div class="comment-pictures-container">
                        <?php
                            try{

```

```
    $commentImages =
getImgagesForComment($comment['id']);
} catch(PDOException $ex){
    http_response_code(500);
    $errorMessageArray[] = 'Server error:
' . $ex->getMessage();
    echo json_encode($errorMessageArray);
}

?>
<?php foreach($commentImages as $image): ?>
    
    <?php endforeach?>
        </div>
    </div>
</article>
<?php endforeach?>
</div>

</section>
```

Checkout.php

```
<?php
    require_once "models/global/functions.php";
    require_once "models/user/functions.php";
    require_once "models/delivery/functions.php";
    require_once "models/book/functions.php";

    if(!isset($_SESSION['userId'])){
        http_response_code(401);
        redirect("index.php?page=login");
        exit;
    }

    try{
        $allDeliveryTypes = selectAllDeliveryTypes();
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
    }

    $deliveryTypes = $allDeliveryTypes->activeDeliveryTypes;

    $userId = $_SESSION['userId'];

    try{
        $userInformationObj = getUserInformation($userId);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
    }

    $addressLine = isset($userInformationObj->address_line) ?
$userInformationObj->address_line : '';
    $addressNumber =isset($userInformationObj->address_number) ?
$userInformationObj->address_number : '';
    $city =isset($userInformationObj->city) ? $userInformationObj-
>city : '';
    $state =isset($userInformationObj->state) ? $userInformationObj-
>state : '';
    $zipCode =isset($userInformationObj->zip_code) ?
$userInformationObj->zip_code : '';
```

```

    $country =isset($userInformationObj->country) ?
$userInformationObj->country : '';

    $cart =json_decode($_COOKIE['cart']) ;

    $totalPrice = 0;
?>

<title>Checkout | Readily</title>
</head>
<body>

<section id="cart-items" class="wrapper">
    <div id="table-container">

        <table>
            <thead>
                <tr>
                    <td class="td-image-checkout">Image</td>
                    <td class="td-title">Title</td>
                    <td class="td-unit-price">Unit Price</td>
                    <td class="td-quantity">Quantity</td>
                    <td class="td-price">Price</td>
                    <td class="td-remove">Remove</td>
                </tr>
            </thead>
            <tbody id="tbody-checkout">
                <?php foreach($cart as $bookInCart):?>
                    <?php $bookInformation =
selectBook($bookInCart->id); ?>

                    <tr id="tr-<?=$bookInCart->id ?>">
                        <td class="td-image">
                            ">
                        </td>
                        <td class="td-title">
                            <div>
                                <p><?=$bookInformation['title'] ?><
/p>
                            </div>
                        </td>
                        <td class="td-unit-price">
                            <p>$<?=$bookInformation['price'] ?>.00<
/p>
                        </td>
                        <td class="td-quantity">

```

```


<div>
        <input min="0" type="number"
value="<?=$bookInCart->quantity ?>" class="input-qunatity"
id="quantity-<?=$bookInCart->id ?>">
    </div>
</td>
<td class="td-price">
    <p id="price-107">$
    <?php
        $price = $bookInformation['price']
* $bookInCart->quantity;
        $totalPrice += $price;
        echo $price;
    ?>.00</p>
</td>
<td class="td-remove">
    <div>
        <i class="fa-solid fa-xmark"
id="remove-<?=$bookInCart->id ?>">
            </i>
    </div>
</td>
</tr>

<?php endforeach?>
</tbody>
</table>

<div id="empty" style="display:<?php
if(!count($cart)){
    echo 'block';
} else{
    echo 'none';
}
?>">
    <p>The Cart Is Empty</p>
</div>

</div>

<div id="total-price-container">
    <p id="total-price">Total: $<?=$totalPrice ?>.00</p>
</div>
</section>


```

```

<section class="sign-up-container wrapper" id="sign-up-form">

    <form id="delivery-form" method="post" name="delivery-form"
action="form.php">

        <h2>Address info</h2>

        <div id="checkout-address-container" class="info-
container-grid address-info-container-grid">

            <label for="address-line-input-js" class="input-title
address-line-title">Address line</label>
                <input type="text" value=<?=$addressLine ?>" id="address-line-input-js" class="address-line-input" name="address-
line-input">
                    <p id="address-line-error" class="error-message " style="display: none;">Please enter a valid address line.</p>

            <label for="number-input-js" class="input-title number-
title">Number</label>
                <input type="text" value=<?=$addressNumber ?>" id="number-input-js" class="number-input" name="number-input">
                    <p id="number-error" class="error-message " style="display: none;">Please enter a valid address number.</p>

            <label for="delivery-input" class="input-title delivery-
title">Delivery option</label>
                <select name="delivery-input" id="delivery-input">
                    <?php foreach($deliveryTypes as $deliveryType): ?>
                        <option value=<?=$deliveryType['id'] ?>><?=$
deliveryType['name'] ?></option>
                    <?php endforeach?>
                </select>
                <p id="delivery-error" class="error-message " style="display: none;">Please enter a valid delivery type option.</p>

            <label for="city-input-js" class="input-title city-
title">City</label>
                <input type="text" value=<?=$city ?>" id="city-input-
js" class="city-input" name="city-input">
                    <p id="city-error" class="error-message " style="display: none;">Incorrect format (ex. Los Angeles)</p>

            <label for="state-input-js" class="input-title state-
title">State</label>
                <input type="text" value=<?=$state ?>" id="state-
input-js" class="state-input" name="state-input">

```

```
        <p id="state-error" class="error-message" style="display: none;">Incorrect format (ex. California)</p>

        <label for="zip-code-input-js" class="input-title zip-code-title">Zip-code</label>
            <input type="text" value=<?=$zipCode ?>" id="zip-code-input-js" class="zip-code-input" name="zip-code-input">
                <p id="zip-code-error" class="error-message" style="display: none;">Incorrect format (ex. 90001)</p>

        <label for="country-input-js" class="input-title country-title">Country</label>
            <input type="text" value=<?=$country ?>" id="country-input-js" class="country-input" name="country-input">
                <p id="country-error" class="error-message" style="display: none;">Incorrect format (ex. United States)</p>
            </div>

        <div class="server-messages">
        </div>

        <input type="button" id="create-account-button" class="checkout-button" value="Submit">

    </form>
</section>
```

Home.php

```
<?php
    require_once "models/home/functions.php";
    require_once "models/book/functions.php";
    require_once "models/category/functions.php";

    $surveyWebsiteExperience = [];

    if(isset($_SESSION['userId'])){
        try{
            $surveyWebsiteExperience = getSurvey('website
experience',$_SESSION['userId']);
        }catch(PDOException $ex){
            http_response_code(500);
            $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
            echo json_encode($errorMessageArray);
        }
    }

    try{
        $bestSellingBooks = getBestSellingBooks();
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
    }

    try{
        $allParentCategories = selectAllActiveCategories(null);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
    }

    try{
        $allCategories = getCategories($allParentCategories);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
    }

    $peferedCategories = [];
```

```

if(isset($_SESSION['userId'])){
    try{
        $peferedCategories =
selectUserPreferredCategories($_SESSION['userId']);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
    }

    if(count($peferedCategories)){
        try{
            $preferedBooks =
selectRelatedBooksToACategories($peferedCategories);
        }catch(PDOException $ex){
            http_response_code(500);
            $errorMessageArray[] = 'Server error: ' . $ex-
>getMessage();
            echo json_encode($errorMessageArray);
        }

        shuffle($preferedBooks);
    }
}
?>

<title>Home | Readily</title>
</head>
<body>

    <section id="about-readily">
        <h1>What is Readily?</h1>
        <div id="about-container">
            <p>At Readily, we're passionate about providing you
with the books you love. With our online bookstore, you can easily
browse and purchase your favorite books, all from the comfort of your
own home.</p>
        </div>
        <div id="img-placeholder">
            
        </div>
    </section>

    <div id="angle-down">
        <div id="position"></div>

```

```

        <a href="#position">
            <i id="scroll-down" class="fa-solid fa-angle-down"></i>
        </a>

    </div>
    <section id="introduction" class="wrapper">
        <div id="introduction-heading-container">
            <h2>Whatever you're into, we've got <br> something for
you</h2>
        </div>

        <article id="article1">
            
            <h3>Discover something new every day</h3>
            <p>At Readily, we're all about discovering something new.
With our vast selection of titles and new releases added regularly,
there's always something fresh and exciting to explore. Join us today
and start your journey of discovery.</p>
        </article>

        <article id="article2">
            
            <h3>Originals you won't find elsewhere</h3>
            <p>At Readily, we pride ourselves on offering a selection
of books that you won't find anywhere else. From rare editions to
limited releases, we're passionate about bringing you unique and
original titles that will inspire and delight you. Join us today and
discover the joy of owning a one-of-a-kind book.</p>
        </article>

        <article id="article3">
            
            <h3>Help finding your next great read</h3>
            <p>At Readily, we're here to help you find your next
favorite book. Whether you're looking for a specific genre, author, or
just need some inspiration, our knowledgeable staff are on hand to
offer recommendations and advice. With our user-friendly website and
personalized service, discovering your next great read has never been
easier. Join us today and let us help you find your next literary
obsession.</p>
        </article>
    </section>

    <section id="discover">

```

```

        <div id="discover-container">
            <h2>Read anywhere. Anytime.</h2>
            <p>Discover the best reads on various topics of
interest.</p>
            <a href="index.php?page=shop">Explore titles</a>
        </div>
    </section>

    <section id="suggested-books" class="wrapper section-articles">
        <?php if(count($peferedCategories)): ?>
            <h2 class="section-heading">Books Recommended For You</h2>
            <div class="article-container article-book-container"
id="sugested-books-articles-container">
                <?php foreach($preferredBooks as $book): ?>

                    <?=displayBookArticle($book) ?>

                <?php endforeach ?>
            </div>

        <?php endif; ?>
    </section>

    <section id="bestselling-books" class="wrapper section-articles">
        <h2 class="section-heading">Bestselling Books</h2>

        <div class="article-container article-book-container"
id="bestselling-books-articles-container">
            <?php foreach($bestSellingBooks as $book): ?>

                <?=displayBookArticle($book) ?>

            <?php endforeach ?>
        </div>
    </section>

    <section id="popular-categories" class="wrapper section-articles">
        <h2 class="section-heading">Popular Categories</h2>

        <div class="article-container article-category-container"
id="popular-categories-articles-container">
            <?php foreach($allCategories as $category): ?>

```

```

<?=displayCategoryArticle($category) ?>

    <?php endforeach ?>
</div>
</section>

<?php if(count($surveyWebsiteExperience)): ?>

<section id="survey-website-experience-section" class="wrapper">

    <h2><?= $surveyWebsiteExperience[0]['name'] ?></h2>

    <form id="website-experience-form">
        <?php foreach($surveyWebsiteExperience as $option): ?>
            <article class="survey-option">
                ">
                <p><?=$option['choice'] ?></p>
                <input type="radio" name="survey-choice" hidden value="<?=$option['id'] ?>">
                <input type="text" name="survey-id" hidden value="<?=$option['survey_id'] ?>">
            </article>
        <?php endforeach ?>
    </form>

    <div class="server-messages survey-message" style="display: none;">
        </div>
    </section>

    <?php endif ?>

    <?php if(isset($_SESSION['userId'])):>

        <form class="sign-up-container" id="message-form" >

            <h2>Get in Touch with Us</h2>

            <div class="info-container-grid personal-info-container-grid">

                <label for="subject-input-js" class="input-title first-name-title">Subject</label>
                <input type="text" id="subject-input-js" class="first-name-input" name="subject">

```

```
<p id="first-name-error" class="error-message "
style="display: none;">Can't be empty</p>

    <label for="biography-input-js" id="biography-title-js"
class="input-title biography-title" >Message</label>
        <textarea name="message-input" id="biography-input-js"
class="biography-input" ></textarea>
            <p id="biography-error" class="error-message "
style="display:none">There must be at least 5 words.</p>

    </div>

    <div class="server-messages" style="display:none">
    </div>

    <div class="button-option-container">
        <input type="button" class="safe-option" id="send-
message-form" value="Send">
    </div>
</form>
<?php endif?>
```

Insert-book.php

```
<?php
    require_once "config/connection.php";
    require_once "models/global/functions.php";
    require_once "models/book/functions.php";
    require_once "models/publisher/functions.php";
    require_once "models/category/functions.php";

    if(!isset($_SESSION["userRoleName"])) ||
    (($_SESSION["userRoleName"] != "admin") && $_SESSION["userRoleName"] != "writer") || !isset($_GET['writer-id'])){
        http_response_code(403);
        redirect("index.php?page=home");
        exit;
    }

    $writerId = $_GET['writer-id'];

    try{
        $allPublishers = selectAllPublishers();
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
    }

    $activePublishers = $allPublishers->activePublishers;

    try{
        $allCategories = selectAllActiveCategories(null);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
    }
?>

<title>Insert Book | Readily</title>
</head>
<body>

<section class="insert-update-book-section wrapper">
```

```

        <form class="book-form-container" id="insert-book-form"
enctype="multipart/form-data" name="insert-book-form" action=""
method="post">
    <input type="text" name="writer-id" value="<?=$writerId?>" hidden>
    <h2>Book Info</h2>

    <div class="book-image-container-form">
    </div>

    <p class="upload-book-image-text">upload image of the book</p>

    <input type="file" id="book-image-js" name="book-image"
class="book-image-form"/>
    <p id="book-image-error" class="error-message "
style="display:none">Please upload an image with a maximum size of
700KB and in one of the following formats: jpg, jpeg, or png. Image
width needs to be smaller than its height</p>

    <div class="info-container-grid book-info-container-grid">

        <label for="book-title-title-js" class="input-title book-title-
title">Book Title</label>
        <input type="text" id="book-title-title-js" class="book-title-
input" name="book-title-input">
        <p id="book-title-error" class="error-message " style="display:
none;">Can't be empty</p>

        <label for="page-count-input-js" class="input-title page-count-
title">Page Count</label>
        <input type="number" min="0" id="page-count-input-js"
class="page-count-input" name="page-count-input">
        <p id="page-count-error" class="error-message " style="display:
none;">Can't be empty</p>

        <label for="price-input-js" class="input-title price-
title">Price</label>
        <input type="number" min="0" id="price-input-js" class="price-
input" name="price-input">
        <p id="price-error" class="error-message " style="display:
none;">Can't be empty</p>

        <label for="release-date-input-js" class="input-title release-
date-title">Release Date</label>
        <input type="date" id="release-date-input-js" class="release-
date-input" name="release-date-input">
        <p id="release-date-error" class="error-message "
style="display: none;">Can't be empty and can't be in future</p>

```

```

        <label for="publisher-input-js" class="input-title publisher-
title">Publisher</label>
        <select id="publisher-input-js" name="publisher-input"
class="publisher-input">
            <?php foreach($activePublishers as $publisher): ?>
            <option
value="<?= $publisher['id']?>" ><?= $publisher[ 'name' ]?></option>
            <?php endforeach ?>
        </select>
        <p id="publisher-error" class="error-message " style="display:
none;">Can't be empty</p>

        <label for="book-description-input-js" class="input-title
book-description-title" >Description</label>
        <textarea name="book-description-input" id="book-description-
input-js" class="book-description-input" ></textarea>
        <p id="book-description-error" class="error-message "
style="display: none;">Can't be empty</p>

        <p class="input-title book-category-title">Categories</p>
        <ul class="book-category-input" id="book-category-input-js">
            <?php foreach($allCategories as $category): ?>
            <li class="category-parent"><p><i class="fa-solid fa-
angle-down"></i><?= $category[ 'name' ]?></p>
                <?= displayCategoriesMenu($category[ 'id' ],0)?>
            </li>
            <?php endforeach ?>
        </ul>
        <p id="book-category-error" class="error-message "
style="display: none;">Can't be empty</p>
    </div>

    <div class="server-messages">
    </div>

    <div class="button-option-container">

        <input type="button" class="safe-option" id="insert-new-book"
value="Add Book">
    </div>
</form>

</section>

```

Login.php

```
<?php
    require_once "models/global/functions.php";

    if(isset($_SESSION['userRoleName'])){
        http_response_code(403);
        redirect('index.php?page=home');
        exit;
    }
?>

<title>Log in | Readily</title>
</head>
<body >

<section id="log-in-section">
    <form name="login-form" id="login-form" action="" method="post">

        <h2>LOG IN</h2>

        <div id="log-in-container">
            <label for="username-input-js" id="username-title"
class="input-title">Username</label>
            <input type="text" name="username-input" id="username-
input-js" >
            <p id="username-error" class="error-message" style="display:
none;">Incorrect format</p>

            <label for="password-input-js" id="password-
title" class="input-title">Password</label>
            <input type="password" name="password-input" id="password-
input-js" >
            <p id="password-error" class="error-message" style="display:
none;">Incorrect format</p>

        </div>

        <div class="server-messages"></div>

        <input type="button" value="Log In" id="log-in-button">

    </form>
</section>
```

Logout.php

```
<?php
    require_once "models/global/functions.php";

    session_destroy();

    destroyAllCookies();

    http_response_code(302);
    redirect('index.php?page=home');
?>
```

Shop.php

```
<?php
    require_once "models/category/functions.php";

    try{
        $allCategories = selectAllActiveCategories(null);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
    }

    $categoryId = 0;

    if(isset($_GET['category-id'])){
        $categoryId = $_GET['category-id'];
    }
?>

<title>Shop | Readily</title>
</head>
<body>

<section id="section-container" class="wrapper">

    <form id="filter-sort-section">
        <div id="input-search-container">
            <input type="text" placeholder="Search" name="input-search" id="input-search">
            <i class="fa-solid fa-magnifying-glass"></i>
        </div>

        <h3 id="heading-price">Price</h3>
        <div id="price-container">
            <input type="number" min="0" placeholder="Min" id="price-min" name="price-min">
            <input type="number" min="0" placeholder="Max" id="price-max" name="price-max">
        </div>

        <h3 id="heading-categories">Categories</h3>
        <ul class="book-category-input" id="book-category-input">
js    categories-option">
            <?php foreach($allCategories as $category): ?>
```

```

        <li class="category-parent"><p><i class="fa-solid fa-angle-down"></i><?= $category[ 'name' ]?></p>
            <?= displayCategoriesMenu($category[ 'id' ],$categoryId)?>
        </li>
    <?php endforeach ?>
</ul>

<h3 id="heading-sort">Sort</h3>
<select name="sort" id="sort">
    <option class="option-tag" value="popular">Most Popular</option>
    <option class="option-tag" value="newest">Newest</option>
    <option class="option-tag" value="price-asc">Price Asc</option>
    <option class="option-tag" value="price-desc">Price Desc</option>
    <option class="option-tag" value="name-asc">Name A-Z</option>
    <option class="option-tag" value="name-desc">Name Z-A</option>
</select>

</form>

<section id="section-articles">
    <ul id="articles-container-ul">

    </ul>

    <nav class="pagination-container" aria-label="Page navigation">
        <ul class="pagination" id="pagination-ul">
        </ul>
    </nav>

    <div class="server-messages">
    </div>
</section>

</section>
```

Signup.php

```
<?php
    require_once "models/global/functions.php";

    if(isset($_SESSION['userRoleName'])){
        http_response_code(403);
        redirect("index.php?page=home");
        exit;
    }
?>

<title>Sign up | Readily</title>
</head>
<body id="sign-up-body">

<form class="sign-up-container" id="sign-up-form"
enctype="multipart/form-data" name="signup-form" action=""
method="post">

    <h2>Personal info</h2>

    <div id="user-avatar-placeholder-js" class="user-avatar-
placeholder">
        
    </div>
    <p class="upload-avatar-text">upload avatar</p>

    <input type="file" id="user-avatar-js" name="user-avatar"
class="user-avatar"/>
    <p id="avatar-error" class="error-message "
style="display:none">Please upload an image with a maximum size of
700KB and in one of the following formats: jpg, jpeg, or png for your
avatar.</p>

    <div class="info-container-grid personal-info-container-grid">

        <label for="first-name-input-js" class="input-title first-name-
title">First Name</label>
        <input type="text" id="first-name-input-js" class="first-name-
input" name="first-name-input">
        <p id="first-name-error" class="error-message " style="display:
none;">Incorrect format (ex. Jhon)</p>
```

```

        <label for="last-name-input-js" class="input-title last-name-
title">Last Name</label>
            <input type="text" id="last-name-input-js" class="last-name-
input" name="last-name-input">
                <p id="last-name-error" class="error-message " style="display:
none;">Incorrect format (ex. Smith)</p>

        <label for="username-input-js" class="input-title username-
title">Username</label>
            <input type="text" id="username-input-js" class="username-
input" name="username-input">
                <p id="username-error" class="error-message " style="display:
none;">Your username must be at least 5 characters long and can only
contain letters, numbers, periods, parentheses, forward slashes,
hyphens, and underscores.</p>

        <label for="password-input-js" class="input-title password-
title">Password</label>
            <input type="password" id="password-input-js" class="password-
input" name="password-input">
                <p id="password-error" class="error-message" style="display:
none;">Your password must be at least 5 characters long and contain at
least one uppercase letter, one lowercase letter, one digit, and one
special character from the set of periods, parentheses, forward slashes,
hyphens, and underscores (' . ', '_', '-', '/', '()' ).</p>

        <label for="email-input-js" class="input-title email-
title">Email</label>
            <input type="email" id="email-input-js" class="email-input"
name="email-input">
                <p id="email-error" class="error-message " style="display:
none;">Incorrect format (ex. jhonsmith@gmail.com)</p>

        <label for="phone-input-js" class="input-title phone-
title">Phone</label>
            <input type="number" id="phone-input-js" class="phone-input"
name="phone-input">
                <p id="phone-error" class="error-message " style="display:
none;">Incorrect format (ex. 0611234567)</p>

        <label for="role-input-js" class="input-title role-
title">Role</label>
            <select id="role-input-js" name="role-input" class="role-
input">
                <option value="2" selected>Customer</option>
                <option value="3">Writer</option>
            </select>

```

```

        <p id="role-error" class="error-message" style="display:none;">Role doesn't exist</p>

        <label for="biography-input" id="biography-title-js" class="input-title biography-title" style="display:none;">Biography</label>
            <textarea name="biography-input" id="biography-input-js" class="biography-input" style="display:none;"></textarea>
            <p id="biography-error" class="error-message" style="display:none;">There must be at least 5 words.</p>
        </div>

        <h2>Address info <br /><span>(optional)</span></h2>

        <div class="info-container-grid address-info-container-grid">

            <label for="address-line-input-js" class="input-title address-line-title">Address line</label>
                <input type="text" id="address-line-input-js" class="address-line-input" name="address-line-input">
                <p id="address-line-error" class="error-message" style="display: none;">Please enter a valid address line.</p>

            <label for="number-input-js" class="input-title number-title">Number</label>
                <input type="text" id="number-input-js" class="number-input" name="number-input">
                <p id="number-error" class="error-message" style="display:none;">Please enter a valid address number.</p>

            <label for="city-input-js" class="input-title city-title">City</label>
                <input type="text" id="city-input-js" class="city-input" name="city-input">
                <p id="city-error" class="error-message" style="display:none;">Incorrect format (ex. Los Angeles)</p>

            <label for="state-input-js" class="input-title state-title">State</label>
                <input type="text" id="state-input-js" class="state-input" name="state-input">
                <p id="state-error" class="error-message" style="display:none;">Incorrect format (ex. California)</p>

            <label for="zip-code-input-js" class="input-title zip-code-title">Zip-code</label>
                <input type="text" id="zip-code-input-js" class="zip-code-input" name="zip-code-input">

```

```
<p id="zip-code-error" class="error-message " style="display:  
none;">Incorrect format (ex. 90001)</p>  
  
    <label for="country-input-js" class="input-title country-  
title">Country</label>  
        <input type="text" id="country-input-js" class="country-input"  
name="country-input">  
        <p id="country-error" class="error-message " style="display:  
none;">Incorrect format (ex. United States)</p>  
    </div>  
  
    <div class="server-messages">  
    </div>  
  
    <input type="button" id="create-account-button" value="Create  
Account">  
  
</form>
```

Update-book.php

```
<?php
    require_once "config/connection.php";
    require_once "models/global/functions.php";
    require_once "models/book/functions.php";
    require_once "models/writer/functions.php";
    require_once "models/publisher/functions.php";
    require_once "models/category/functions.php";

    if(!isset($_SESSION["userRoleName"]) ||
    ($_SESSION["userRoleName"] != "admin") && $_SESSION["userRoleName"] != "writer") || !isset($_GET['book-id'])){
        http_response_code(401);
        redirect("index.php?page=login");
        exit;
    }

    $bookId = $_GET['book-id'];

    try{
        $allPublishers = selectAllPublishers();
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
    }

    $activePublishers = $allPublishers->activePublishers;

    try{
        $allCategories = selectAllActiveCategories(null);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
    }

    try{
        $bookInformation = selectBookInformation($bookId);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
    }
```

```

$timestamp = strtotime($bookInformation->release_date);
$releaseDate = date('Y-m-d',$timestamp );
?>

<title>Update Book | Readily</title>
</head>
<body>

<section class="insert-update-book-section wrapper">

<form class="book-form-container" id="update-book-form"
enctype="multipart/form-data" name="update-book-form" action=""
method="post">
<input type="text" id="book-id-js" name="book-id"
value=<?=$bookInformation->id?>" hidden>
<input type="text" id="writer-id-js" name="writer-id"
value=<?=$bookInformation->writer_id?>" hidden>
<h2>Book Info</h2>

<div class="book-image-container-form">
    name ?>">
</div>

<p class="upload-book-image-text">upload image of the book</p>

<input type="file" id="book-image-js" name="book-image" class="book-
image-form"/>
<p id="book-image-error" class="error-message "
style="display:none">Please upload an image with a maximum size of
700KB and in one of the following formats: jpg, jpeg, or png for your
avatar. Image width needs to be smaller than its height</p>

<div class="info-container-grid book-info-container-grid">

    <label for="book-title-title-js" class="input-title book-title-
title">Book Title</label>
    <input type="text" id="book-title-title-js" class="book-title-
input" name="book-title-input" value=<?=$bookInformation->name ?>">
    <p id="book-title-error" class="error-message " style="display:
none;">Can't be empty</p>

    <label for="page-count-input-js" class="input-title page-count-
title">Page Count</label>
    <input type="number" min="0" id="page-count-input-js" class="page-
count-input" name="page-count-input" value=<?=$bookInformation-
>page_count ?>">

```

```

        <p id="page-count-error" class="error-message " style="display: none;">Can't be empty</p>

        <label for="price-input-js" class="input-title price-title">Price</label>
        <input type="number" min="0" id="price-input-js" class="price-input" name="price-input" value=<?=$bookInformation->price ?>>
        <p id="price-error" class="error-message " style="display: none;">Can't be empty</p>

        <label for="release-date-input-js" class="input-title release-date-title">Release Date</label>
        <input type="date" id="release-date-input-js" class="release-date-input" name="release-date-input" value=<?=$releaseDate?>>
        <p id="release-date-error" class="error-message " style="display: none;">Can't be empty and can't be in future</p>

        <label for="publisher-input-js" class="input-title publisher-title">Publisher</label>
        <select id="publisher-input-js" name="publisher-input" class="publisher-input">
            <?php foreach($activePublishers as $publisher): ?>
                <option value=<?=$publisher['id']?><?php if($publisher['id'] == $bookInformation->publisher_id){ echo 'selected'; } ?> ><?=$publisher['name']?></option>
            <?php endforeach ?>
        </select>
        <p id="publisher-error" class="error-message " style="display: none;">Can't be empty</p>

        <label for="book-description-input-js" class="input-title book-description-title" >Description</label>
        <textarea name="book-description-input" id="book-description-input-js" class="book-description-input" ><?=$bookInformation->description ?></textarea>
        <p id="book-description-error" class="error-message " style="display: none;">Can't be empty</p>

        <p class="input-title book-category-title">Categories</p>
        <ul class="book-category-input" id="book-category-input-js">
            <?php foreach($allCategories as $category): ?>
                <li class="category-parent"><p><i class="fa-solid fa-angle-down"></i><?=$category['name']?></p>
                    <?=displayCategoriesMenu($category['id'],0)?>
                </li>
            <?php endforeach ?>
        </ul>
    
```

```
    <p id="book-category-error" class="error-message " style="display:  
none;">Can't be empty</p>  
</div>  
  
<div class="server-messages">  
</div>  
  
<div class="button-option-container">  
    <input type="button" class="safe-option" id="update-book"  
value="Save">  
</div>  
  
</form>  
  
</section>
```

User-profile.php

```
<?php
    require_once "models/global/functions.php";
    require_once "models/user/functions.php";

    if(!isset($_SESSION['userRoleName'])){
        http_response_code(401);
        redirect('index.php?page=login');
        exit;
    }

    $userId = $_SESSION['userId'];

    if($_SESSION['userRoleName'] == 'admin'){
        if(isset($_GET['selected-user'])){
            $userId = $_GET['selected-user'];
        }
    }

    try{
        $userInformationObj = getUserInformation($userId);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
    }

    $userId = $userInformationObj->id;
    $firstName = $userInformationObj->first_name;
    $lastName = $userInformationObj->last_name;
    $username = $userInformationObj->username;
    $email = $userInformationObj->email;
    $phone = $userInformationObj->phone;
    $roleId = $userInformationObj->role_id;
    $roleName = $userInformationObj->role_name;
    $biographyText = $userInformationObj->biography_text;

    $addressLine = isset($userInformationObj->address_line) ?
$userInformationObj->address_line : '/';
    $addressNumber =isset($userInformationObj->address_number) ?
$userInformationObj->address_number : '/';
    $city =isset($userInformationObj->city) ? $userInformationObj-
>city : '/';
```

```

        $state = isset($userInformationObj->state) ? $userInformationObj-
>state : '/';
        $zipCode = isset($userInformationObj->zip_code) ?
$userInformationObj->zip_code : '/';
        $country = isset($userInformationObj->country) ?
$userInformationObj->country : '/';

        $avatarId = isset($userInformationObj->avatar_id) ?
$userInformationObj->avatar_id : '/';
        $avatarSrc = isset($userInformationObj->avatar_src) ?
$userInformationObj->avatar_src : 'default-avatar.jpg';

        setUserInformationInCookie();
?>

<title>Profile | Readily</title>
</head>
<body>

<section id="user-profile-section">

    <div id="user-data" class="sign-up-container">
        <h2>Personal info</h2>

        <div id="user-avatar-placeholder-js" class="user-avatar-
placeholder">
            
        </div>

        <div class="info-container-grid personal-info-container-grid">

            <p class="input-title first-name-title">First Name</p>
            <p class="first-name-input user-
information" ><?=$firstName ?></p>

            <p class="input-title last-name-title">Last Name</p>
            <p class="last-name-input user-
information" ><?=$lastName ?></p>

            <p class="input-title username-title">Username</p>
            <p class="username-input user-
information" ><?=$username ?></p>

            <p class="input-title email-title">Email</p>
            <p class="email-input user-information" ><?=$email ?></p>

            <p class="input-title phone-title">Phone</p>
        
```

```

<p class="phone-input user-information"><?=$phone ?></p>

<p class="input-title role-title">Role</p>
<p class="role-input user-
information"><?=ucfirst($rolename) ?></p>

<p class="input-title biography-title" <?php
if(!isset($biographyText)) echo
'style="display:none;"' ?> >Biography</p>
<p class="user-information biography-input" <?php
if(!isset($biographyText)) echo
'style="display:none;"' ?> ><?=$biographyText ?></p>

</div>

<h2>Address info</h2>

<div class="info-container-grid address-info-container-grid">

<p class="input-title address-line-title">Address line</p>
<p class="address-line-input user-
information" ><?=$addressLine ?></p>

<p class="input-title number-title">Number</p>
<p class="number-input user-
information" ><?=$addressNumber ?></p>

<p class="input-title city-title">City</p>
<p class="city-input user-information" ><?=$city ?></p>

<p class="input-title state-title">State</p>
<p class="state-input user-information" ><?=$state ?></p>

<p class="input-title zip-code-title">Zip-code</p>
<p class="zip-code-input user-
information" ><?=$zipCode ?></p>

<p class="input-title country-title">Country</p>
<p class="user-information country-
input" ><?=$country ?></p>
</div>

<div class="server-messages success-server-messages">
```

```

        </div>

        <div class="button-option-container">
            <input type="button" class="danger-option" id="delete-account-
button" value="Delete">
            <input type="button" class="safe-option" id="update-account-
button" value="Update">
        </div>
    </div>

    <?php
        $addressLine = isset($userInformationObj->address_line) ?
$userInformationObj->address_line : '';
        $addressNumber =isset($userInformationObj->address_number) ?
$userInformationObj->address_number : '';
        $city =isset($userInformationObj->city) ? $userInformationObj-
>city : '';
        $state =isset($userInformationObj->state) ?
$userInformationObj->state : '';
        $zipCode =isset($userInformationObj->zip_code) ?
$userInformationObj->zip_code : '';
        $country =isset($userInformationObj->country) ?
$userInformationObj->country : '';
    ?>

        <form style="display:none;" class="sign-up-container" id="update-
user-form" enctype="multipart/form-data" name="signup-form" action="" 
method="post">

            <!-- password stoji tu samo kako bi funkcija koja brise errore
radila kako treba -->
            <p id="password-error" style="display: none;"></p>
            <input type="text" name="role-input" hidden id="role-input"
value="<?=$roleId ?>">
            <input type="text" name="user-id" hidden id="user-id"
value="<?=$userId ?>">
            <input type="text" name="previous-address" hidden
value="<?=$addressLine ?>">
            <input type="text" name="role-name" hidden value="<?=$roleName ?>">

            <h2>Personal info</h2>

            <div id="user-avatar-placeholder-js" class="user-avatar-
placeholder">
                
            </div>
            <p class="upload-avatar-text">upload avatar</p>

```

```

        <input type="file" id="user-avatar-js" name="user-avatar"
class="user-avatar"/>
        <p id="avatar-error" class="error-message "
style="display:none">Please upload an image with a maximum size of
700KB and in one of the following formats: jpg, jpeg, or png for your
avatar.</p>

        <div class="info-container-grid personal-info-container-grid">

            <label for="first-name-input-js" class="input-title first-name-
title">First Name</label>
            <input type="text" id="first-name-input-js" class="first-name-
input" value=<?=$firstName ?>" name="first-name-input">
            <p id="first-name-error" class="error-message " style="display:
none;">Incorrect format (ex. Jhon)</p>

            <label for="last-name-input-js" class="input-title last-name-
title">Last Name</label>
            <input type="text" id="last-name-input-js" class="last-name-
input" value=<?=$lastName ?>" name="last-name-input">
            <p id="last-name-error" class="error-message " style="display:
none;">Incorrect format (ex. Smith)</p>

            <label for="username-input-js" class="input-title username-
title">Username</label>
            <input type="text" id="username-input-js" class="username-
input" value=<?=$username ?>" name="username-input">
            <p id="username-error" class="error-message " style="display:
none;">Your username must be at least 5 characters long and can only
contain letters, numbers, periods, parentheses, forward slashes,
hyphens, and underscores.</p>

            <label for="email-input-js" class="input-title email-
title">Email</label>
            <input type="email" id="email-input-js" class="email-input"
value=<?=$email ?>" name="email-input">
            <p id="email-error" class="error-message " style="display:
none;">Incorrect format (ex. jhonsmith@gmail.com)</p>

            <label for="phone-input-js" class="input-title phone-
title">Phone</label>
            <input type="number" id="phone-input-js" class="phone-input"
value=<?=$phone ?>" name="phone-input">
            <p id="phone-error" class="error-message " style="display:
none;">Incorrect format (ex. 0611234567)</p>

            <p class="input-title role-title">Role</p>

```

```

<p id="role-input-js" class="role-input user-information"><?=ucfirst($roleName) ?></p>

    <label for="biography-input-js" id="biography-title-js" class="input-title biography-title" <?php if(!isset($biographyText)) echo 'style="display:none;"' ?>Biography</label>
        <textarea name="biography-input" id="biography-input-js" class="biography-input" <?php if(!isset($biographyText)) echo 'style="display:none;"' ?> ><?=$biographyText ?></textarea>
        <p id="biography-error" class="error-message" style="display: none;">There must be at least 5 words.</p>
    </div>

<h2>Address info <br /><span>(optional)</span></h2>

<div class="info-container-grid address-info-container-grid">

    <label for="address-line-input-js" class="input-title address-line-title">Address line</label>
        <input type="text" id="address-line-input-js" class="address-line-input" value="<?=$addressLine ?>" name="address-line-input">
        <p id="address-line-error" class="error-message" style="display: none;">Please enter a valid address line.</p>

    <label for="number-input-js" class="input-title number-title">Number</label>
        <input type="text" id="number-input-js" class="number-input" value="<?=$addressNumber ?>" name="number-input">
        <p id="number-error" class="error-message" style="display: none;">Please enter a valid address number.</p>

    <label for="city-input-js" class="input-title city-title">City</label>
        <input type="text" id="city-input-js" class="city-input" value="<?=$city ?>" name="city-input">
        <p id="city-error" class="error-message" style="display: none;">Incorrect format (ex. Los Angeles)</p>

    <label for="state-input-js" class="input-title state-title">State</label>
        <input type="text" id="state-input-js" class="state-input" value="<?=$state ?>" name="state-input">
        <p id="state-error" class="error-message" style="display: none;">Incorrect format (ex. California)</p>

    <label for="zip-code-input-js" class="input-title zip-code-title">Zip-code</label>

```

```

        <input type="text" id="zip-code-input-js" class="zip-code-
input" value=<?=$zipCode ?>" name="zip-code-input">
            <p id="zip-code-error" class="error-message " style="display:
none;">Incorrect format (ex. 90001)</p>

            <label for="country-input-js" class="input-title country-
title">Country</label>
                <input type="text" id="country-input-js" class="country-input"
value=<?=$country ?>" name="country-input">
                    <p id="country-error" class="error-message " style="display:
none;">Incorrect format (ex. United States)</p>
            </div>

            <div class="server-messages error-server-messages">
            </div>

            <div class="button-option-container">
                <input type="button" class="danger-option" id="cancel-account-
button" value="Cancel">
                <input type="button" class="safe-option" id="save-account-
button" value="Save">
            </div>

        </form>
</section>
<?php if($roleName=='writer'): ?>
    <div id="writer-option-wrapper">
        <div class="admin-option-container">
            <div class="admin-option">
                <a href="index.php?page=writer-books&writer-
id=<?=$userId?>"></a>
                    <div class="option-name-container">
                        <p>Books</p>
                    </div>
                    <div class="option-icon-container">
                        
                    </div>
            </div>
        </div>
    </div>
<?php endif ?>
```

Writer-books.php

```
<?php
    require_once "config/connection.php";
    require_once "models/global/functions.php";
    require_once "models/writer/functions.php";

    if(!isset($_SESSION["userRoleName"]) ||
    ($_SESSION["userRoleName"] != "admin") && $_SESSION["userRoleName"] != "writer") ){
        http_response_code(401);
        redirect("index.php?page=login");
        exit;
    }

    $writerId = $_GET['writer-id'];

    try{
        $booksTable = selectAllWriterBooksForTable($writerId);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
    }
?>

<title>Writer Books | Readily</title>
</head>
<body>

<section id="writer-books-section" class="wrapper">

    <h2>Active Books</h2>

    <div class="table-container">
        <table id="table-active-books">
            <thead>
                <tr>
                    <th>Image</th>
                    <th>Title</th>
                    <th>Price</th>
                    <th>Release Date</th>
                    <th>Update</th>
                    <th>Delete</th>
                </tr>
            <tbody>
```

```

        </thead>
        <tbody>
            <?php foreach ($booksTable->activeBooks as $book): ?>
            <tr>
                <td class="td-image">"></td>
                <td><?=$book['name']?></td>
                <td>$<?=$book['price']?>.00</td>
                <td><?=$book['release_date']?></td>
                <td><a class="safe-option change-links"
href="index.php?page=update-book&book-
id=<?=$book['id'] ?>">Change</a></td>
                <td>
                    <form method="post" name="delivery-form" >
                        <input type="text"
value="<?=$book['id'] ?>" hidden name="book-id">
                        <input type="text"
value="<?=$book['writer_id'] ?>" hidden name="writer-id">
                        <input class="danger-option delete-
book" type="button" value="Delete">
                    </form>
                </td>
            </tr>
            <?php endforeach; ?>
        </tbody>
    </table>
</div>
<div class="add-button-container add-buttons-container" >
    <a class="safe-option" href="index.php?page=insert-book&writer-
id=<?=$writerId?>">Add Book</a>
</div>

<h2>Deleted Books</h2>

<div class="table-container">
    <table id="table-deleted-books">
        <thead>
            <tr>
                <th>Image</th>
                <th>Title</th>
                <th>Price</th>
                <th>Deleted At</th>
                <th>Update</th>
                <th>Activate</th>
            </tr>
        </thead>
        <tbody>

```

```

<?php foreach ($booksTable->deletedBooks as $book): ?>
    <tr>
        <td class="td-image">"></td>
        <td><?=$book['name']?></td>
        <td>$<?=$book['price']?>.00</td>
        <td><?=$book['release_date']?></td>
        <td><a class="safe-option change-links"
href="index.php?page=update-book&book-
id=<?=$book['id'] ?>">Change</a></td>
        <td>
            <form method="post" name="delivery-form" >
                <input type="text"
value="<?=$book['id'] ?>" hidden name="book-id">
                <input type="text"
value="<?=$book['writer_id'] ?>" hidden name="writer-id">
                <input class="safe-option activate-
book" type="button" value="Activate">
            </form>
        </td>
    </tr>
<?php endforeach; ?>
</tbody>
</table>
</div>
</section>

```

Writer.php

```
<?php
    require_once "models/global/functions.php";
    require_once "models/writer/functions.php";
    require_once "models/book/functions.php";
    require_once "models/category/functions.php";

    if(!isset($_GET['id'])){
        http_response_code(400);
        redirect("index.php?page=home");
        exit;
    }

    $writerId = $_GET['id'];

    try{
        $writerInformation = selectWriter($writerId);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
    }

    $writerInformation['src'] = $writerInformation['src'] ?
$writerInformation['src'] : "default-avatar.jpg";

    $words = explode(' ', $writerInformation['biography_text']);
    $shortBiography = array_slice($words, 0, 25);
    $shortBiography = implode(' ', $shortBiography);
    $shortBiography .= "... ";

    try{
        $allWritersBooks = selectWriterBooks($writerId);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
    }

    try{
        $writerCategories = selectWriterCategories($writerId);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
```

```

        echo json_encode($errorMessageArray);
    }

    try{
        $categoriesInformation = getCategories($writerCategories);
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
    }

    try{
        $allWriters = selectAllWriters();
    }catch(PDOException $ex){
        http_response_code(500);
        $errorMessageArray[] = 'Server error: ' . $ex->getMessage();
        echo json_encode($errorMessageArray);
    }
?>

<title>Writer | Readily</title>
</head>
<body>

<section id="writer-info">
    <div id="writer-info-container" class="wrapper">
        <h1><?= $writerInformation['writer']; ?></h1>
        
        <p class="short-info" style="display:
block;"><?= $shortBiography ?><span id="show-more">Show
More</span></p><p class="long-info" style="display:
none;"><?= $writerInformation['biography_text'] ?><span id="show-
less">Show Less</span></p></div>
</section>

<section id="writer-books" class="wrapper section-articles">
    <h2 class="section-heading">Books by J.K. Rowling</h2>
    <div class="article-container article-book-container" id="writer-
books-articles-container">
        <?php foreach($allWritersBooks as $book): ?>

            <?= displayBookArticle($book) ?>

            <?php endforeach ?>
    </div>
</section>

```

```
<section id="writer-categories" class="wrapper section-articles" >
    <h2 class="section-heading">Related Categories</h2>
    <div class="article-container article-category-container"
id="writer-categories-articles-container">
        <?php foreach($categoriesInformation as $category): ?>

            <?=displayCategoryArticle($category) ?>

        <?php endforeach ?>
    </div>
</section>

<section id="other-writers" class="wrapper section-articles" >
    <h2 class="section-heading">Check Out Other Authors</h2>
    <div class="article-container article-author-container" id="other-
writers-articles-container">
        <?php foreach($allWriters as $writer): ?>

            <?=displayWriterArticle($writer) ?>

        <?php endforeach ?>
    </div>
</section>
```

Index.php

```
<?php
session_start();
ob_start();
require_once "config/connection.php";

include "views/fixed/head.php";
include "views/fixed/header.php";

$pages = ["home", "shop", "checkout", "writer", "book", "signup",
"login", "logout", "user-profile", "admin-visitors", "author",
"404", "writer-books", "insert-book", "update-book", "admin", "admin-
users", "admin-categories", "admin-publishers", "admin-delivery-
options", "admin-survey", "admin-messages", "admin-orders", "admin-
logged-id-users"];

if(isset($_GET['page'])){
    if(in_array($_GET['page'],$pages)){
        include "views/pages/" . $_GET['page'] . ".php";
    }else{
        include "views/pages/404.php";
    }
}else{
    include "views/pages/home.php";
}

include "views/fixed/footer.php";
ob_end_flush();
?>
```

Main.js

```
var url = window.location.search;
var prefersDarkScheme;

console.log(url);

window.onload =function(){
    setThemeClass();
    setHeader();

    if(url.includes("page=signup")){
        console.log("signup stranica")
        setSignupForm();
    }

    if(url.includes("page=login")){
        console.log("login stranica")
        setLoginForm();
    }

    if(url.includes("page=user-profile")){
        console.log("user-profile stranica")
        setUserAccount();
        setUpdateUserForm();
    }

    if(url.includes("page=admin-users")){
        console.log("admin-users stranica")
        setTableZebra();
        setEventsForUserTable();
    }

    if(url.includes("page=admin-publishers")){
        console.log("admin-publishers stranica")
    }
}
```

```

        setEventsForPublisherPage();
        setTableZebra();
        setEventsForPublisherTable();
    }

    if(url.includes("page=admin-survey")){
        console.log("admin-survey stranica")

        setTableZebra();
        setEventsForSurveyTable();
    }

    if(url.includes("page=admin-categories")){
        console.log("admin-categories stranica")

        setEventsForCategoriesPage();
        setTableZebra();
        setEventsForCategoriesTable();
    }

    if(url.includes("page=admin-messages")){
        console.log("admin-messages stranica")

        setTableZebra();
    }

    if(url.includes("page=admin-orders")){
        console.log("admin-orders stranica")

        setTableZebra();
    }

    if(url.includes("page=admin-delivery-options")){
        console.log("admin-delivery-options stranica")

        setEventsForDeliveryTypesPage();
        setTableZebra();
        setEventsForDeliveryTypesTable();
    }

    if(url.includes("page=writer-books")){
        console.log("writer-books stranica")
    }
}

```

```

        setEventsForWriterBooksPage();
        setTableZebra();
    }

    if(url.includes("page=insert-book")){
        console.log("insert-book stranica")
        setInsertBookForm();
    }

    if(url.includes("page=update-book")){
        console.log("update-book stranica")
        selectBookCategories();
        setUpdateBookForm();
    }

    if(url.includes("page=book")){
        console.log("book stranica")

        createAngleFunctionality('related-categories-
container','.article-category','book-related-categories-
section','category');
        createAngleFunctionality('div-info-section','.article-info-
container','div-info-container','category')
        createAngleFunctionality('related-books','#related-books-
articles-container','related-books-articles-container','books')
        createAngleFunctionality('authors-books','#autors-books-
articles-container','autors-books-articles-container','books')

        let formReview = document.getElementById('star-rating-form');

        if(formReview){
            let starsContainer = document.querySelectorAll('.star-icon-
container');

            starsContainer.forEach(star=>{
                star.addEventListener('click',()=>{
                    let radioBtn =
star.querySelector('input[type="radio"]');
                    radioBtn.checked = true;
                    let rating = radioBtn.value;

                    let stars = document.querySelectorAll('.book-
rating-star');

```

```

        for(let i=0;i<stars.length;i++){
            if(rating>=i){
                stars[i].classList.remove('fa-regular');
                stars[i].classList.add('fa-solid');

            }
            else{
                stars[i].classList.remove('fa-solid');
                stars[i].classList.add('fa-regular');
            }
        }

        let formData = new FormData(formReview);

        $.ajax({
            url: 'models/book/book-rating.php',
            type: 'POST',
            data: formData,
            contentType: false,
            processData: false,
            success: function(response) {

                displayServerMessages('server-
messages',response,'success-message');
            },
            error: function(xhr, status, errorThrown) {
                console.log('neuspeh')
                let messages = JSON.parse(xhr.responseText);
                displayServerMessages('server-
messages',messages,'error-message');

                console.log(messages)
                console.log(xhr);
                console.log(status, errorThrown);
            }
        });
    });

}

let formComment = document.getElementById('form-comment');

if(formComment){

    $(document).on('blur','#biography-input-js',checkBiography);

    let inputTag = document.querySelector('.comment-image');
    commentImageInput(inputTag);
}

```

```

$(document).on('click', '#comment-button', sendCommentForm);

}

}

if(url.includes('page=home') || !url.includes("page")){
    console.log("home stranica")

    setMessageForm();

    createAngleFunctionality('bestselling-books', '.article-book', 'bestselling-books-articles-container', 'books');
    createAngleFunctionality('popular-categories', '.article-category', 'popular-categories-articles-container', 'category');

    let surveyContainer = document.getElementById('survey-website-experience-section');

    if(surveyContainer){
        $('.survey-option').on('click', function() {
            var closestRadio = $(this).closest('.survey-option').find('input[type=radio]');
            closestRadio.prop('checked', true);

            sendSurvey('website-experience-form');
        });
    }

    let cookie = document.cookie.split(';');
    for(let i=0; i<cookie.length;i++){

        let name = cookie[i].trim().split('=')[0];
        let value =
decodeURIComponent(cookie[i].trim().split('=')[1]);

        if(name == 'hasPreferredCategories'){
            createAngleFunctionality('suggested-books', '#suggested-books', 'suggested-books-articles-container', 'books');
        }
    }

    window.addEventListener("scroll", ()=>{
        let element = document.getElementById("position");
        let scrollPosition = window.scrollY;

```

```

        let positionOffset = element.getBoundingClientRect().top +
scrollPosition;

        if(scrollPosition < positionOffset){
            return;
        }

        let cookie = document.cookie.split(';');

        let hasPreferredCategories = false;
        let isLoggedIn = false;

        for(let i=0; i<cookie.length;i++){

            let name = cookie[i].trim().split('=')[0];
            let value =
decodeURIComponent(cookie[i].trim().split('=')[1]);

            switch(name){
                case 'hasPreferredCategories':
                    hasPreferredCategories = value;
                    break;
                case 'isLoggedIn':
                    isLoggedIn = value;
                    break;
            }
        }

        if(hasPreferredCategories || !isLoggedIn){
            return;
        }

        let exist = document.querySelector(".choosing-categories-
section");
        if(exist != undefined){
            return;
        }

        $("body").addClass("pre-loader");
        let body = document.querySelector("body");

        let bodyChildren = body.children;
        for(let i=0;i<bodyChildren.length;i++){
            bodyChildren[i].classList.add("blur");
        }

        let titleTag = document.createElement("h2");

```

```

        titleTag.innerText = "Select 3 categories that match your
interests"

        let choosingCategoriesSection =
document.createElement("section");
        let choosingCategoriesContainer =
document.createElement("form");
        choosingCategoriesContainer.setAttribute('id','user-
preferred-categories-form');

        choosingCategoriesContainer.appendChild(titleTag);

        choosingCategoriesSection.classList.add("choosing-
categories-section");
        choosingCategoriesContainer.classList.add("choosing-
categories-container");

        displayChoosingCategories(choosingCategoriesContainer);

        choosingCategoriesSection.appendChild(choosingCategoriesCon
tainer);
        body.appendChild(choosingCategoriesSection);
    })
}

if(url.includes("page=writer&id")){
    console.log("writer stranica")

    createAngleFunctionality('writer-books','.article-
book','writer-books-articles-container','books');
    createAngleFunctionality('other-writers','.article-
writer','other-writers-articles-container','author');

    $(document).on('click','#show-more',()=>{
        let short = document.querySelector(".short-info");
        let long = document.querySelector(".long-info");

        short.style.display = "none";
        long.style.display = "block";
    })

    $(document).on('click','#show-less',()=>{
        let short = document.querySelector(".short-info");
        let long = document.querySelector(".long-info");

        long.style.display = "none";
        short.style.display = "block";
    })
}

```

```

}

if(url.includes("page=shop")){
    console.log("shop stranica")
    setFilterSortForm();
    sendFilterSortForm();
}

if(url.includes("page=checkout")){
    console.log("checkout stranica")
    setTableZebra();
    setCheckoutTableEvents();
    setCheckoutForm();
}

if(url.includes("page=admin-logged-in-users")){
    setTableZebra();
}

if(url.includes("page=admin-visitors")){
    setTableZebra();
}

let cartTag = document.querySelectorAll('.fa-cart-shopping');

cartTag.forEach(element=>{
    element.addEventListener('click',()=>{
        let string = element.getAttribute('id');
        let id = parseInt(string.match(/\d+$/)[0]);
        addToCart(id);

    })
})
}

function sendCommentForm(){
    let errorCount = 0;

    errorCount += checkBiography();

    errorCount += checkCommentImages();
    if(errorCount){
        return;
    }
}

```

```

}

let form = document.getElementById('form-comment');
let formData = new FormData(form);

$.ajax({
    url: 'models/book/insert-comment.php',
    type: 'POST',
    data: formData,
    contentType: false,
    processData: false,
    success: function(response) {

        document.getElementById('biography-input-js').value = "";

        let commentImages = document.querySelectorAll('.comment-image');
        let commentImagesArray =
Array.prototype.slice.call(commentImages, 1);

        commentImages.forEach((elem) => {
            elem.value = '';
        });

        commentImagesArray.forEach((elem) => {
            elem.remove();
        });

        displayServerMessages('comment-server-error',response,'success-message');
    },
    error: function(xhr, status, errorThrown) {
        let messages = JSON.parse(xhr.responseText);
        displayServerMessages('comment-server-error',messages,'error-message');

        console.log(messages)
        console.log(xhr);
        console.log(status, errorThrown);
    }
});

function checkCommentImages(){
    let commentImages = document.querySelectorAll('.comment-image');
    let errorMessage = document.getElementById('comment-images-error');

    let totalSize = 0;
}

```

```

let isValid = true;

Array.from(commentImages).forEach(image => {

    if (image.files.length > 0) {
        let file = image.files[0];
        let fileType = file.type.split('/')[1];
        let fileSize = file.size / (1024 * 1024);

        if (fileType !== 'jpg' && fileType !== 'jpeg' && fileType !==
'png') {
            isValid = false
        }

        totalSize += fileSize;

        if (totalSize > 2) {
            isValid = false
        }
    }
});

if(isValid){
    errorMessage.style.display = "none";
    return 0;
}

errorMessage.style.display = "block";
return 0;
}

function commentImageInput(element){

    element.addEventListener('change',()=>{
        let commentImagesContainer = document.querySelector('#comment-
images-container');
        let commentImages = document.querySelectorAll('.comment-image');

        let uploadedImages = Array.from(commentImages).filter(image =>
image.files.length > 0);
        checkCommentImages();
        if (element.files.length > 0 && commentImages.length < 3 ){
            let newInput = document.createElement('input');
            newInput.type = 'file';
            newInput.name = 'comment-image[]';
            newInput.classList.add('comment-image');
            commentImageInput(newInput);
        }
    })
}

```

```

        commentImagesContainer.appendChild(newInput);
    }else if(element.files.length == 0 && uploadedImages.length
<2){
    if (commentImages.length > 1) {
        element.remove();
    }
}
})

}

function setMessageForm(){
    $(document).on('blur','#biography-input-js',checkBiography);

    $(document).on('blur','#subject-input-js',()=>{
        let subject = document.getElementById('subject-input-js').value;
        errorMessage = document.getElementById('first-name-error');

        if(!subject.length){
            errorMessage.style.display = 'block';
            return 1;
        }

        errorMessage.style.display = 'none';
        return 0;
    });

    $(document).on('click','#send-message-form',sendMessageForm);
}

function sendMessageForm(){
    let errorCount = 0;

    let subject = document.getElementById('subject-input-js').value;
    errorMessage = document.getElementById('first-name-error');

    if(!subject.length){
        errorMessage.style.display = 'block';
        errorCount++;
    }
    errorCount += checkBiography();

    if(errorCount){
        return;
    }

    let form = document.getElementById('message-form');

```



```

        checkInputElementWithRegex(/^(A-ZŠĐĆČŽ][a-
zšđććž]{2,}\s?)+$/,'state-input-js','state-error');
    });
    $(document).on('blur','#country-input-js',()=>{
        checkInputElementWithRegex(/^(A-ZŠĐĆČŽ][a-
zšđććž]{2,}\s?)+$/,'country-input-js','country-error');
    });
    $(document).on('blur','#zip-code-input-js',()=>{
        checkInputElementWithRegex(/^d{5,15}$/, 'zip-code-input-
js','zip-code-error');
    });

    $(document).on('click','.checkout-button',sendCheckoutData);
}

function sendCheckoutData(){
    let errorCount = 0;

    errorCount += checkInputElementWithRegex(/^a-zšđććA-ZŠĐĆČ0-
9\s\.\-\#\!\/\,\,]+$/,'address-line-input-js','address-line-error');
    errorCount += checkInputElementWithRegex(/^d+[a-zA-Z]?$/,'number-
input-js','number-error');
    errorCount += checkInputElementWithRegex(/^(A-ZŠĐĆČ][a-
zšđććž]{2,}\s?)+$/,'city-input-js','city-error');
    errorCount += checkInputElementWithRegex(/^(A-ZŠĐĆČŽ][a-
zšđććž]{2,}\s?)+$/,'state-input-js','state-error');
    errorCount += checkInputElementWithRegex(/^(A-ZŠĐĆČŽ][a-
zšđććž]{2,}\s?)+$/,'country-input-js','country-error');
    errorCount += checkInputElementWithRegex(/^d{5,15}$/, 'zip-code-
input-js','zip-code-error');

    let cookieValue =
document.cookie.replace(/(?:^|.*;\s*)cart\s*\=\s*([^\;]*).*$/|^.*$/,
"$1");
    cookieValue = decodeURIComponent(cookieValue.replace(/\+/g, ' '));
    let cart = JSON.parse(cookieValue);

    if(!cart.length){
        errorCount++;
        let message = ['Your cart is empty. Please add a book to
proceed with the purchase.'];
        displayServerMessages('server-messages',message,'error-
message');
    }

    if(errorCount){
        return;
    }
}

```

```

let form = document.getElementById('delivery-form');
let formData = new FormData(form);

$.ajax({
    url: 'models/order/order-logic.php',
    type: 'POST',
    data: formData,
    contentType: false,
    processData: false,
    success: function(response) {
        clearAllInputValuesCheckout();
        clearAllInputErrorsCheckout();

        displayServerMessages('server-messages', response, 'success-
message');
    },
    error: function(xhr, status, errorThrown) {
        let messages = JSON.parse(xhr.responseText);
        displayServerMessages('server-messages', messages, 'error-
message');

        console.log(messages)
        console.log(xhr);
        console.log(status, errorThrown);
    }
});

function clearAllInputErrorsCheckout(){
    document.getElementById('address-line-error').style.display =
'none';
    document.getElementById('number-error').style.display = 'none';
    document.getElementById('delivery-error').style.display = 'none';
    document.getElementById('city-error').style.display = 'none';
    document.getElementById('state-error').style.display = 'none';
    document.getElementById('zip-code-error').style.display = 'none';
    document.getElementById('country-error').style.display = 'none';

    let messagesContainer = document.querySelector('.server-messages');
    while(messagesContainer.firstChild)messagesContainer.removeChild(me
ssagesContainer.firstChild);
}

function clearAllInputValuesCheckout(){
    displayUserCartTable([]);

    document.getElementById('address-line-input-js').value = '';
    document.getElementById('number-input-js').value = '';
}

```

```

document.getElementById('city-input-js').value = '';
document.getElementById('state-input-js').value = '';
document.getElementById('zip-code-input-js').value = '';
document.getElementById('country-input-js').value = '';
}

function setCheckoutTableEvents(){
    let removeTags = document.querySelectorAll('.fa-xmark');

    removeTags.forEach(tag=>{
        tag.addEventListener('click',()=>{
            let tagId = tag.getAttribute('id')
            let idBook = parseInt(tagId.match(/\d+$/)[0]);
            console.log(idBook)

            let cookieValue =
document.cookie.replace(/(?:^(?:^|.*;\s*)cart\s*\=\s*([^\;]*).*$/,'$1');
            cookieValue = decodeURIComponent(cookieValue.replace(/\+/g,
' '));
            let cart = JSON.parse(cookieValue);

            for(let i=0;i<cart.length;i++){
                if(cart[i].id == idBook){
                    cart.splice(i,1);
                }
            }

            let jsonString = JSON.stringify(cart);
            document.cookie = `cart=${jsonString};path=/`;

            $.ajax({
                url: 'models/cart/cart-logic.php',
                type: 'POST',
                data: {},
                contentType: false,
                processData: false,
                success: function(response) {
                    displayUserCartTable(response);
                    setCheckoutTableEvents();
                    setTableZebra();
                    console.log(" ")
                    console.log("PHP:")
                    console.log(response);
                },
                error: function(xhr, status, errorThrown) {
                    let messages = JSON.parse(xhr.responseText);

```

```

        console.log(messages)
        console.log(xhr);
        console.log(status, errorThrown);
    }
})
})
}

let inputTags = document.querySelectorAll('.input-qunatity');

inputTags.forEach(tag=>{
    tag.addEventListener('change', ()=>{
        let tagId = tag.getAttribute('id')
        console.log('id ' + tagId)
        let idBook = parseInt(tagId.match(/\d+$/)[0]);
        let quantityValue = document.getElementById(`quantity-
${idBook}`).value;

        let cookieValue =
document.cookie.replace(/(?:^(?:^|.*;\s*)cart\s*\=\s*([^\;]*).*$)|^.*$/,
"$1");
        cookieValue = decodeURIComponent(cookieValue.replace(/\+/g,
' '));
        let cart = JSON.parse(cookieValue);

        for(let i=0;i<cart.length;i++){
            if(cart[i].id == idBook){
                if(Number(quantityValue) != 0){
                    cart[i].quantity = quantityValue;
                }else{
                    cart.splice(i,1);
                }
            }
        }

        let jsonString = JSON.stringify(cart);
        document.cookie = `cart=${jsonString};path=/`;

        $.ajax({
            url: 'models/cart/cart-logic.php',
            type: 'POST',
            data: {},
            contentType: false,
            processData: false,
            success: function(response) {
                displayUserCartTable(response);
                setCheckoutTableEvents();
            }
        });
    })
});
}

```

```

        setTableZebra();
        console.log(" ")
        console.log("PHP:")
        console.log(response);
    },
    error: function(xhr, status, errorThrown) {
        let messages = JSON.parse(xhr.responseText);

        console.log(messages)
        console.log(xhr);
        console.log(status, errorThrown);
    }
});

}

function displayUserCartTable(books){
    let tbodyElement = document.getElementById('tbody-checkout');

    while(tbodyElement.firstChild)tbodyElement.removeChild(tbodyElement.firstChild);

    if(books.length == 0){
        let trEmptyTag = document.getElementById("empty");
        trEmptyTag.style.display = "block"
    }else{

        let trEmptyTag = document.getElementById("empty");
        //console.log(trEmptyTag);
        trEmptyTag.style.display = "none";
    }

    let totalPrice = 0;

    for(let i=0;i<books.length;i++){
        let book = books[i];
        let trTag = document.createElement("tr");
        trTag.setAttribute("id", `tr-${book.id}`);

        let imgTag = document.createElement("img");
        imgTag.classList.add("set-brightness");
        imgTag.src = `assets/images/books/small/${book.src}`;
        imgTag.alt = book.title;

        let tdTagImg = document.createElement("td")
        tdTagImg.classList.add("td-image");

```

```

        tdTagImg.appendChild(imgTag);

        let titleTag = document.createElement("p");
        titleTag.innerText = book.title;

        let titleContainer = document.createElement("div");
        titleContainer.appendChild(titleTag);

        let tdTitle = document.createElement("td");
        tdTitle.classList.add("td-title")
        tdTitle.appendChild(titleContainer)

        let unitPriceTag = document.createElement("p");
        unitPriceTag.innerText = `$$ {book.price}.00`;

        let tdUnitPriceTag = document.createElement("td");
        tdUnitPriceTag.classList.add("td-unit-price")
        tdUnitPriceTag.appendChild(unitPriceTag);

        let quantityTag = document.createElement("input");
        quantityTag.classList.add('input-quanity');
        quantityTag.setAttribute("min", "0");
        quantityTag.type = "number";
        quantityTag.setAttribute("id", `quantity-{book.id}`);
        quantityTag.value = `${book.quantity}`;
        let quantityTagContainer = document.createElement("div");

        quantityTagContainer.appendChild(quantityTag);

        let tdQuantity = document.createElement("td");
        tdQuantity.classList.add("td-quantity");
        tdQuantity.appendChild(quantityTagContainer)

        let priceTag = document.createElement("p");
        priceTag.setAttribute("id", `price-{book.id}`);
        priceTag.innerText = "$ " + (book.price
*book.quantity).toFixed(2);
        totalPrice += book.price *book.quantity;

        let tdPrice = document.createElement("td");
        tdPrice.classList.add("td-price");
        tdPrice.appendChild(priceTag)

        let removeTag = document.createElement("i");
        removeTag.classList.add("fa-solid", "fa-xmark");
        removeTag.setAttribute("id", `remove-{book.id}`);

        let removeTagContainer = document.createElement("div");
        removeTagContainer.appendChild(removeTag);

```

```

let tdRemove = document.createElement("td");
tdRemove.classList.add("td-remove");

tdRemove.appendChild(removeTagContainer);

trTag.appendChild(tdTagImg);
trTag.appendChild(tdTitle);
trTag.appendChild(tdUnitPriceTag);
trTag.appendChild(tdQuantity);
trTag.appendChild(tdPrice);
trTag.appendChild(tdRemove);

tbodyElement.appendChild(trTag);

}

let totalPriceTag = document.getElementById('total-price');
totalPriceTag.textContent = `$$ {totalPrice}.00`;
}

function sendFilterSortForm(activePage = 1){
    let form = document.getElementById('filter-sort-section');

    let formData = new FormData(form);
    formData.append('activePage', activePage);

    $.ajax({
        url: 'models/shop/shop.php',
        type: 'POST',
        data: formData,
        contentType: false,
        processData: false,
        success: function(response) {
            let element = document.querySelector('.server-messages');
            element.style.display = 'none';
            displayShopBooks(response[2]);
            createPageNumbers(response[0], response[1]);

            if(!(response[2].length)){
                element.style.display = 'block';
                element.innerHTML = `<p class="error-message">We're
sorry, but we couldn't find any books that match your interests.</p>`;
            }
        },
        error: function(xhr, status, errorThrown) {
            let messages = JSON.parse(xhr.responseText);
            let element = document.querySelector('.server-messages');
            element.style.display = 'block';
        }
    });
}

```

```

        displayServerMessages('server-messages',messages,'error-
message');

        console.log(messages)
        console.log(xhr);
        console.log(status, errorThrown);
    }
});

}

function createPageNumbers(pageCount,activePage){
    let ulTag = document.getElementById('pagination-ul');

    while(ulTag.firstChild)ulTag.removeChild(ulTag.firstChild);

    for(let i=1;i<=pageCount;i++){
        let liTag = document.createElement('li');
        liTag.classList.add('page-item','number');

        if(i == activePage){
            liTag.classList.add('active');
        }

        let aTag = document.createElement('a');
        aTag.setAttribute('href','');
        aTag.classList.add('page-link');
        aTag.textContent = i;

        aTag.addEventListener('click',(i)=>{
            sendFilterSortForm(aTag.textContent)
        })

        liTag.appendChild(aTag);
        ulTag.appendChild(liTag);
    }
}

function displayShopBooks(books){

    let ulTag = document.getElementById('articles-container-ul');
    while(ulTag.firstChild)ulTag.removeChild(ulTag.firstChild);

    books.forEach(book=>{
        let liTag = document.createElement('li');
        liTag.classList.add('li-tag-article-container')
        let article = bookArticleGenerator(book,"",true);
        liTag.appendChild(article);
    })
}

```

```

        ulTag.appendChild(liTag);
    })
}

function setFilterSortForm(){

    let parentCategories = document.querySelectorAll('.category-parent
ul');

    parentCategories.forEach(parentCat=>{
        parentCat.style.display = 'none';
    })

    $('.category-parent').on('click', function() {
        let nestedUl = $(this).find('ul');
        nestedUl.toggle();
    });

    $('.book-category-cb').on('click', function(event) {
        event.stopPropagation();
    });

    $('.book-category-li label').on('click', function(event) {
        event.stopPropagation();
    });

    $(document).on('blur', '#input-search', ()=>{
        sendFilterSortForm();
    })

    $(document).on('blur', '#price-min', ()=>{
        sendFilterSortForm();
    })

    $(document).on('blur', '#price-max', ()=>{
        sendFilterSortForm();
    })

    $(document).on('change', '#sort', ()=>{
        sendFilterSortForm();
    })

    let cbElements = document.querySelectorAll('.book-category-cb');

    cbElements.forEach(element=>{
        element.addEventListener('change', ()=>{
            sendFilterSortForm();
        })
    })
}

```

```

        })
    }

function setEventsForSurveyTable(){
    let deleteSurveys = document.querySelectorAll('.delete-survey');
    deleteSurveys.forEach(deleteSurvey => {
        deleteSurvey.addEventListener('click', () => {
            let form = deleteSurvey.closest('form');
            sendDeleteSurveyForm(form);
        });
    });

    let activatePublishers = document.querySelectorAll('.activate-survey');
    activatePublishers.forEach(activatePublisher => {
        activatePublisher.addEventListener('click', () => {
            let form = activatePublisher.closest('form');
            sendActivateSurveyForm(form);
        });
    });
}

function sendActivateSurveyForm(form){
    let formData = new FormData(form);

    $.ajax({
        url: 'models/survey/activate-survey.php',
        type: 'POST',
        data: formData,
        contentType: false,
        processData: false,
        success: function(response) {
            populateSurveyTables(response);
        },
        error: function(xhr, status, errorThrown) {

            let messages = JSON.parse(xhr.responseText);

            console.log(messages)
            console.log(xhr);
            console.log(status, errorThrown);
        }
    });
}

function sendDeleteSurveyForm(form){
    let formData = new FormData(form);

```

```

$.ajax({
    url: 'models/survey/delete-survey.php',
    type: 'POST',
    data: formData,
    contentType: false,
    processData: false,
    success: function(response) {
        populateSurveyTables(response);
    },
    error: function(xhr, status, errorThrown) {
        let messages = JSON.parse(xhr.responseText);

        console.log(messages)
        console.log(xhr);
        console.log(status, errorThrown);
    }
});

function populateSurveyTables(surveys){
    let activeSurveys = surveys[0].activeSurveys;
    let deletedSurveys = surveys[0].deletedSurveys;

    let acuteSurveysTable = document.querySelector('#table-active-surveys tbody');
    let deletedSurveysTable = document.querySelector('#table-deleted-surveys tbody');

    while(acuteSurveysTable.firstChild)acuteSurveysTable.removeChild(acuteSurveysTable.firstChild);
    while(deletedSurveysTable.firstChild)deletedSurveysTable.removeChild(deletedSurveysTable.firstChild);

    for(let property in activeSurveys){
        let trTag = document.createElement('tr');

        let tdTagName = document.createElement('td');
        let tdTagCreatedAt = document.createElement('td');
        let tdTagDelete = document.createElement('td');

        tdTagName.textContent = activeSurveys[property].name;
        tdTagCreatedAt.textContent = activeSurveys[property].created_at;

        let formDelete = document.createElement('form');

        let inputSurveyIdDelete = document.createElement('input');
        inputSurveyIdDelete.setAttribute('type', 'text');

```

```

inputSurveyIdDelete.setAttribute('name','survey-id');
inputSurveyIdDelete.style.display = 'none';
inputSurveyIdDelete.value = activeSurveys[property].id;

let inputButtonDelete = document.createElement('input');
inputButtonDelete.setAttribute('type','button');
inputButtonDelete.classList.add('danger-option','delete-
survey');
inputButtonDelete.value = 'Delete';

formDelete.appendChild(inputSurveyIdDelete);
formDelete.appendChild(inputButtonDelete);
tdTagDelete.appendChild(formDelete);

trTag.appendChild(tdTagName);
trTag.appendChild(tdTagCreatedAt);

for(let option in activeSurveys[property].options){
    tdTagOption = document.createElement('td');
    tdTagOption.classList.add('survey-table-options');

    nameTag = document.createElement('p');
    valueTag = document.createElement('p');

    nameTag.textContent = option;
    valueTag.textContent =
activeSurveys[property].options[option]

    tdTagOption.appendChild(nameTag);
    tdTagOption.appendChild(valueTag);

    trTag.appendChild(tdTagOption)
}

trTag.appendChild(tdTagDelete);

acuteSurveysTable.appendChild(trTag);
}

for(let property in deletedSurveys){
    let trTag = document.createElement('tr');

    let tdTagName = document.createElement('td');
    let tdTagFinishedAt = document.createElement('td');
    let tdTagDelete = document.createElement('td');

    tdTagName.textContent = deletedSurveys[property].name;

```

```

        tdTagFinishedAt.textContent =
    deletedSurveys[property].finished_at;

        let formDelete = document.createElement('form');

        let inputSurveyIdDelete = document.createElement('input');
        inputSurveyIdDelete.setAttribute('type', 'text');
        inputSurveyIdDelete.setAttribute('name', 'survey-id');
        inputSurveyIdDelete.style.display = 'none';
        inputSurveyIdDelete.value = deletedSurveys[property].id;

        let inputButtonDelete = document.createElement('input');
        inputButtonDelete.setAttribute('type', 'button');
        inputButtonDelete.classList.add('safe-option', 'activate-
survey');
        inputButtonDelete.value = 'Activate';

        formDelete.appendChild(inputSurveyIdDelete);
        formDelete.appendChild(inputButtonDelete);
        tdTagDelete.appendChild(formDelete);

        trTag.appendChild(tdTagName);
        trTag.appendChild(tdTagFinishedAt);

        for(let option in deletedSurveys[property].options){
            tdTagOption = document.createElement('td');
            tdTagOption.classList.add('survey-table-options');

            nameTag = document.createElement('p');
            valueTag = document.createElement('p');

            nameTag.textContent = option;
            valueTag.textContent =
    deletedSurveys[property].options[option]

            tdTagOption.appendChild(nameTag);
            tdTagOption.appendChild(valueTag);

            trTag.appendChild(tdTagOption)
        }

        trTag.appendChild(tdTagDelete);

        deletedSurveysTable.appendChild(trTag);
    }

    setTableZebra();

```

```

        setEventsForSurveyTable();
    }

function sendSurvey(idForm){
    let form = document.getElementById(idForm);

    let formData = new FormData(form);

    $.ajax({
        url: 'models/survey/insert-survey-webisite-experience-data.php',
        type: 'POST',
        data: formData,
        contentType: false,
        processData: false,
        success: function(response) {
            $('#website-experience-form').animate({
                height: 0
            }, 200);

            let element = document.querySelector('.survey-message');
            element.style.display = 'block';
            displayServerMessages('survey-message',response,'success-
message');
        },
        error: function(xhr, status, errorThrown) {
            let messages = JSON.parse(xhr.responseText);
            let element = document.querySelector('.survey-message');
            element.style.display = 'block';
            displayServerMessages('survey-message',messages,'error-
message');

            console.log(messages)
            console.log(xhr);
            console.log(status, errorThrown);
        }
    });
}

function getCategories(){

}

function displayChoosingCategories(element){

    $.ajax({
        url: 'models/category/get-active-categories.php',
        type: 'POST',
        dataType: 'json',

```

```

success:function(allCategories){
    console.log(allCategories);
    for(let i=0;i<allCategories.length;i++){
        let article =
categoryArticleGenerator(allCategories[i]);

        let checkbox = document.createElement("input");
        checkbox.type = "checkbox";
        checkbox.name = "checkbox-prefered-categories[]"
        checkbox.value = allCategories[i].category_id;
        checkbox.classList.add("category-checkbox-choose");
        checkbox.setAttribute("id",`prefered-category-
${allCategories[i].category_id}`)

        $(document).on("change",`#prefered-category-
${allCategories[i].category_id}`,settingUserPreferedCategories);

        article.appendChild(checkbox);
        element.appendChild(article);

    }

},
error: function(xhr, status, errorThrown) {
    let messages = JSON.parse(xhr.responseText);
    let element = document.querySelector('.server-messages');
    element.style.display = 'block';
    displayServerMessages('server-messages',messages,'error-
message');

    console.log(messages)
    console.log(xhr);
    console.log(status, errorThrown);
}
}

function categoryArticleGenerator(category){
    let article = document.createElement("article");

    let name = category.category_name;

    let nameTag = document.createElement('p');
    nameTag.innerText = name;

    let imgTag = document.createElement("img");
    imgTag.src = `assets/images/books/small/${category.src}`;
    imgTag.alt = category.name;
}

```

```

    imgTag.classList.add("set-brightness");

    let link = document.createElement("a");
    link.href = `index.php?page=shop&id=${category.category_id}`;

    article.classList.add("article-category");
    article.appendChild(link);
    article.appendChild(nameTag);
    article.appendChild(imgTag);

    article.classList.add("article-category");

    return article
}

function bookArticleGenerator(book,label,image){

    let writer_id = book.writer_id;

    let article = document.createElement("article");
    article.classList.add("article-book");

    let divImgContainer = document.createElement("div");

    let lastDigit = book.id.toString().slice(-1);
    divImgContainer.classList.add(`bg-article-color-${lastDigit}`);
    divImgContainer.classList.add('article-div-img-container');

    let bookImg = document.createElement("img");
    bookImg.setAttribute("data-id",`${book.id}`);
    bookImg.classList.add("set-brightness");

    if(image){
        bookImg.src = `assets/images/books/small/${book.src}`;
    }
    bookImg.alt = book.title;

    let divTextConatiner = document.createElement("div");
    divTextConatiner.classList.add("article-books-text-container");

    let divTitleAndAuthor = document.createElement("div");
    divTitleAndAuthor.classList.add("title-and-author");

    let titleTag = document.createElement("h3");
    let authorTag = document.createElement("a");
    authorTag.classList.add("author-link");
    authorTag.href = `index.php?page=writer&id=${writer_id}`;
    authorTag.innerText = book.writer;
}

```

```

if(book.title.length >33){
    titleTag.innerText = book.title.substring(0,30) + " ...";
}else{
    titleTag.innerText = book.title;
}

divTitleAndAuthor.appendChild(titleTag);
divTitleAndAuthor.appendChild(authorTag);

let divStarsAndCart = document.createElement("div");
divStarsAndCart.classList.add("stars-and-cart-container");

let cartTag = document.createElement("i");
cartTag.classList.add('fa-solid','fa-cart-shopping','shopping-
cart');
cartTag.setAttribute("id",'book-id-'+book.id);

cartTag.addEventListener("click",()=>{
    addToCart(book.id);
})

let divStarsContainer = document.createElement("div");
divStarsContainer.classList.add("stars-container");
for(let i=0;i<5;i++){
    if(book.review >i){
        divStarsContainer.innerHTML +=`<i class="fa-solid fa-
star"></i>`;
    }else{
        divStarsContainer.innerHTML +=`<i class="fa-regular fa-
star"></i>`;
    }
}

let reviewNumber = document.createElement("p");
reviewNumber.classList.add("rating-text")
if(book.review>0){
    reviewNumber.innerText=`${book.review}/5`;
}else{
    reviewNumber.innerText= "0 ratings";
}

divStarsAndCart.appendChild(cartTag);
divStarsAndCart.appendChild(divStarsContainer);
divStarsAndCart.appendChild(reviewNumber);

divTextContainer.appendChild(divTitleAndAuthor);
divTextContainer.appendChild(divStarsAndCart);

```

```

let linkToABookTag = document.createElement("a");
linkToABookTag.classList.add("link-to-single-a-book");
linkToABookTag.href = `index.php?page=book&id=${book.id}`;

divImgContainer.appendChild(bookImg);
article.appendChild(divImgContainer);
article.appendChild(divTextConatiner);
article.appendChild(linkToABookTag);

return article;
}

function settingUserPreferredCategories(){
    let checkBoxes = document.querySelectorAll(".category-checkbox-choose");

    let preferred = [];
    checkBoxes.forEach(function(checkbox){
        if (checkbox.checked) {
            preferred.push(checkbox.value);
        }
    })

    if(preferred.length == 3){
       .setItemToLocalStorage("hasPreferredCategories",true);
        let bodyChildren = document.querySelectorAll(".blur");
        for(let i=0;i<bodyChildren.length;i++){
            bodyChildren[i].classList.remove("blur");
            bodyChildren[i].classList.add("un-blur");
        }

        let body = document.querySelector("body");
        body.classList.remove("pre-loader")

        let section = document.querySelector(".choosing-categories-section");
        section.style.display = "none";
        displaySuggestions(preferred);
    }
}

function displaySuggestions(){
    let form = document.getElementById('user-preferred-categories-form');

    let formData = new FormData(form);

    $.ajax({

```

```

        url: 'models/book/get-suggested-books.php',
        type: 'POST',
        data: formData,
        contentType: false,
        processData: false,
        success:function(books){
            sectionGenerator("suggested-books","sugested-books-
articles-container","Books Recommended For
You",bookArticleGenerator,books,"books")
        },
        error: function(xhr, status, errorThrown) {
            let messages = JSON.parse(xhr.responseText);
            let element = document.querySelector('.server-messages');
            element.style.display = 'block';
            displayServerMessages('server-messages',messages,'error-
message');
            console.log(messages)
            console.log(xhr);
            console.log(status, errorThrown);
        }
    })
}

function addToCart(id){

    let cookieValue =
document.cookie.replace(/(?:^(?:^|.*;\s*)cart\s*\=\s*([^\;]*).*$|^.*$/,
"\$1");
    cookieValue = decodeURIComponent(cookieValue.replace(/\+/g, ' '));

    if(!cookieValue){
        window.location.href = `index.php?page=login`;
        return;
    }

    let cart = JSON.parse(cookieValue);
    console.log(" ");
    console.log("JS:");
    cart.forEach(car=>{
        console.log(car);
    })

    let item = cart.find(x=>x.id == id);

    if(item == null){
        item = {
            id : id,

```

```

        quantity:1
    }
    cart.push(item);
}else{
    item.quantity++;
}

let jsonString = JSON.stringify(cart);
document.cookie = `cart=${jsonString};path=/;`;

$.ajax({
    url: 'models/cart/cart-logic.php',
    type: 'POST',
    data: {},
    contentType: false,
    processData: false,
    success: function(response) {
        console.log(" ")
        console.log("PHP:")
        console.log(response);
    },
    error: function(xhr, status, errorThrown) {
        let messages = JSON.parse(xhr.responseText);

        console.log(messages)
        console.log(xhr);
        console.log(status, errorThrown);
    }
});

function
sectionGenerator(locationId, setId, heading, callback, data, typeOfArticle){

    let articleWidth;
    let articleGap;

    let section = document.getElementById(locationId);
    let headingTag = document.createElement("h2");

    headingTag.innerText = heading;
    headingTag.classList.add("section-heading");
    let articleContainerTag = document.createElement("div");
    articleContainerTag.classList.add("article-container");
    articleContainerTag.setAttribute("id",setId);
    for(let i=0; i<data.length;i++){
        let article;

        if(typeOfArticle == "books"){

```

```

        article = callback(data[i],false,true);
        articleWidth = 180;
        articleGap = 40;
        articleContainerTag.classList.add("article-book-container");
    }else if(typeOfArticle == "category"){
        article = callback(data[i].id);
        articleWidth = 250;
        articleGap = 20;
        articleContainerTag.classList.add("article-category-
container");
    }else if(typeOfArticle == "info"){
        article = callback(data[i]);
        articleWidth = 220;
        articleGap = 20;
        articleContainerTag.classList.add("article-info-container");
    }
    else{

        let parametar = window.location.href.split('?');
        let nameAndValue = parametar[1].split('=');
        let idSkip = nameAndValue[1];

        if(data[i].id == idSkip)continue;

        article = callback(data[i].id);
        articleWidth = 180;
        articleGap = 40;
        articleContainerTag.classList.add("article-author-
container");
    }

    articleContainerTag.appendChild(article);
}

section.appendChild(headingTag);
section.appendChild(articleContainerTag);

// Adding scrolling functionality

// Creating left and right angle container
let leftAngleDiv = document.createElement("div");
let rightAngleDiv = document.createElement("div");

leftAngleDiv.setAttribute("id",setId + "-left-angle-container");
rightAngleDiv.setAttribute("id",setId + "-right-angle-container");

leftAngleDiv.classList.add("angle-container");
rightAngleDiv.classList.add("angle-container");

```

```

// Adding left and right angle icons
leftAngleDiv.innerHTML = '<i class="fa-solid fa-angle-left"></i>';
rightAngleDiv.innerHTML = '<i class="fa-solid fa-angle-right"></i>';

leftAngleDiv.style.left = '-20px';
rightAngleDiv.style.right = '-20px';

leftAngleDiv.style.bottom = (articleContainerTag.offsetHeight/2 - 20)
+ "px";
rightAngleDiv.style.bottom = (articleContainerTag.offsetHeight/2 -
20) + "px";

section.appendChild(leftAngleDiv);
section.appendChild(rightAngleDiv);

checkAngleVisibility(setId, `${setId}-left-angle-
container`, `${setId}-right-angle-container`);

$(document).on("click", `#${setId}-left-angle-container`, ()=>{
    scrollLeft(setId, articleWidth, articleGap, `${setId}-left-angle-
container`, `${setId}-right-angle-container`);
});

$(document).on("click", `#${setId}-right-angle-container`, ()=>{
    scrollRight(setId, articleWidth, articleGap, `${setId}-left-angle-
container`, `${setId}-right-angle-container`);
});

$(document).on("wheel", `#${setId}` , ()=>{
    checkAngleVisibility(setId, `${setId}-left-angle-
container`, `${setId}-right-angle-container`);
})

window.addEventListener("resize", ()=>{
    checkAngleVisibility(setId, `${setId}-left-angle-
container`, `${setId}-right-angle-container`);
})
}

function createAngleFunctionality(locationId, articleClass, setId,
typeOfArticle){
    let section = document.getElementById(locationId);
    articleContainerTag = document.querySelector(articleClass);

    let articleWidth;
    let articleGap;

    switch (typeOfArticle){

```

```

        case 'books':
            articleWidth = 180;
            articleGap = 40;
            break;
        case 'category':
            articleWidth = 250;
            articleGap = 20;
            break;
        case 'info':
            articleWidth = 220;
            articleGap = 20;
            break;
        case 'author':
            articleWidth = 180;
            articleGap = 40;
            break;
    }

    let leftAngleDiv = document.createElement("div");
    let rightAngleDiv = document.createElement("div");

    leftAngleDiv.setAttribute("id", setId + "-left-angle-container");
    rightAngleDiv.setAttribute("id", setId + "-right-angle-container");

    leftAngleDiv.classList.add("angle-container");
    rightAngleDiv.classList.add("angle-container");

    leftAngleDiv.innerHTML = '<i class="fa-solid fa-angle-left"></i>';
    rightAngleDiv.innerHTML = '<i class="fa-solid fa-angle-right"></i>';

    leftAngleDiv.style.left = '-20px';
    rightAngleDiv.style.right = '-20px';

    leftAngleDiv.style.bottom = (articleContainerTag.offsetHeight/2 - 20)
+ "px";
    rightAngleDiv.style.bottom = (articleContainerTag.offsetHeight/2 - 20) + "px";

    section.appendChild(leftAngleDiv);
    section.appendChild(rightAngleDiv);

    checkAngleVisibility(setId, `${setId}-left-angle-
container`, `${setId}-right-angle-container`);

    $(document).on("click", `#${setId}-left-angle-container`, ()=>{
        scrollLeft(setId, articleWidth, articleGap, `${setId}-left-angle-
container`, `${setId}-right-angle-container`);
    });

```

```

$(document).on("click", `#${setId}-right-angle-container`, ()=>{
    scrollRight(setId, articleWidth, articleGap, `${setId}-left-angle-
container`, `${setId}-right-angle-container`);
});

$(document).on("wheel", `#${setId}` , ()=>{
    checkAngleVisibility(setId, `${setId}-left-angle-
container`, `${setId}-right-angle-container`);
})

window.addEventListener("resize", ()=>{
    checkAngleVisibility(setId, `${setId}-left-angle-
container`, `${setId}-right-angle-container`);
})

}

function scrollRight(id,articleWidth,gap,idLeft,idRight){
    let container = document.getElementById(id);
    let widthOfAnArticle = articleWidth + gap;

    let widthScrolled = Math.ceil(container.scrollLeft/widthOfAnArticle)
* widthOfAnArticle;
    let articlesThatCanFit =
Math.floor(container.clientWidth/widthOfAnArticle) * widthOfAnArticle;

    container.scrollLeft = widthScrolled+ articlesThatCanFit;

    let leftAngle = document.getElementById(idLeft);
    leftAngle.style.display = "grid";
    let rightAngle = document.getElementById(idRight);
    if((container.scrollWidth-container.clientWidth) <= widthScrolled+
articlesThatCanFit){
        rightAngle.style.display = "none";
    }else{
        rightAngle.style.display = "grid";
    }
}

function scrollLeft(id,articleWidth,gap,idLeft,idRight){
    let container = document.getElementById(id);
    let widthOfAnArticle = articleWidth + gap;

    let scrollRightWidth = container.scrollWidth - container.scrollLeft
- container.clientWidth;

```

```

let articlesThatCanFit =
Math.floor(container.clientWidth/widthOfAnArticle) * widthOfAnArticle;
let rightWidthScrolled =
Math.ceil(scrollRightWidth/widthOfAnArticle) * widthOfAnArticle;

    container.scrollLeft =container.scrollWidth - (rightWidthScrolled +
articlesThatCanFit + container.clientWidth) ;

    let rightAngle = document.getElementById(idRight);
rightAngle.style.display = "grid";
let leftAngle = document.getElementById(idLeft);

    if((container.scrollWidth - (rightWidthScrolled +
articlesThatCanFit + container.clientWidth))<=0){
        leftAngle.style.display = "none";
    }else{
        leftAngle.style.display = "grid";
    }
}

function checkAngleVisibility(idElement,idLeft,idRight){
    let container = document.getElementById(idElement);

    let leftAngle = document.getElementById(idLeft);
    let rightAngle = document.getElementById(idRight);

    if(container.scrollLeft == 0){
        leftAngle.style.display = "none";
    }else{
        leftAngle.style.display = "grid";
    }

    if((container.scrollWidth - (container.scrollLeft +
container.clientWidth)) >0 ){
        rightAngle.style.display = "grid";
    }else{
        rightAngle.style.display = "none";
    }
}

function setEventsForWriterBooksPage(){
    let deleteBooks = document.querySelectorAll('.delete-book');
    deleteBooks.forEach(book => {
        book.addEventListener('click', () => {
            let form = book.closest('form');
            sendDeleteBookFormWrter(form);
        });
    });
}

```

```

let activateBooks = document.querySelectorAll('.activate-book');
activateBooks.forEach(book => {
    book.addEventListener('click', () => {
        let form = book.closest('form');
        sendActivateBookFormWriter(form);
    });
});

}

function populateWriterBooksTables(books){
    console.log(books);
    let activeBooks = books[0].activeBooks;
    let deletedBooks = books[0].deletedBooks;

    let activeBooksTable = document.querySelector('#table-active-books
tbody');
    let deletedBooksTable = document.querySelector('#table-deleted-
books tbody');

    while(activeBooksTable.firstChild)activeBooksTable.removeChild(acti
veBooksTable.firstChild);
    while(deletedBooksTable.firstChild)deletedBooksTable.removeChild(de
letedBooksTable.firstChild);

    activeBooks.forEach(book=>{
        let trTag = document.createElement('tr');

        let tdTagImg = document.createElement('td');
        let tdTagTitle = document.createElement('td');
        let tdTagPrice = document.createElement('td');
        let tdTagReleaseDate = document.createElement('td');
        let tdTagChange = document.createElement('td');
        let tdTagDelete = document.createElement('td');

        tdTagImg.classList.add('td-image');

        imgTag = document.createElement('img');
        imgTag.setAttribute('src', `assets/images/books/small/${book.src}
`);

        imgTag.setAttribute('alt', `${book.name}`);
        tdTagImg.appendChild(imgTag);

        tdTagTitle.textContent = book.name;
        tdTagPrice.textContent = '$' + book.price + '.00';
        tdTagReleaseDate.textContent = book.release_date;
    });
}

```

```

let aTag = document.createElement('a');
aTag.classList.add('safe-option','change-links');
aTag.textContent = 'Change';
aTag.setAttribute('href', `index.php?page=update-book&book-
id=${book.id}`);
tdTagChange.appendChild(aTag);

let formDelete = document.createElement('form');

let inputBookIdDelete = document.createElement('input');
inputBookIdDelete.setAttribute('type','text');
inputBookIdDelete.setAttribute('name','book-id');
inputBookIdDelete.style.display = 'none';
inputBookIdDelete.value = book.id;

let inputWriterIdDelete = document.createElement('input');
inputWriterIdDelete.setAttribute('type','text');
inputWriterIdDelete.setAttribute('name','writer-id');
inputWriterIdDelete.setAttribute('value',`${book.writer_id}`);
inputWriterIdDelete.style.display = 'none';
inputWriterIdDelete.value = book.writer_id;

let inputButtonDelete = document.createElement('input');
inputButtonDelete.setAttribute('type','button');
inputButtonDelete.classList.add('danger-option','delete-book');
inputButtonDelete.setAttribute('value',`${book.book_id}`);
inputButtonDelete.value = 'Delete';

formDelete.appendChild(inputBookIdDelete);
formDelete.appendChild(inputWriterIdDelete);
formDelete.appendChild(inputButtonDelete);
tdTagDelete.appendChild(formDelete);

trTag.appendChild(tdTagImg);
trTag.appendChild(tdTagTitle);
trTag.appendChild(tdTagPrice);
trTag.appendChild(tdTagReleaseDate);
trTag.appendChild(tdTagChange);
trTag.appendChild(tdTagDelete);

activeBooksTable.appendChild(trTag);
})

deletedBooks.forEach(book=>{
let trTag = document.createElement('tr');

let tdTagImg = document.createElement('td');
let tdTagTitle = document.createElement('td');

```

```

let tdTagPrice = document.createElement('td');
let tdTagReleaseDate = document.createElement('td');
let tdTagChange = document.createElement('td');
let tdTagActivate = document.createElement('td');

tdTagImg.classList.add('td-image');

imgTag = document.createElement('img');
imgTag.setAttribute('src', `assets/images/books/small/${book.src}`);
);
imgTag.setAttribute('alt', `${book.name}`);

tdTagImg.appendChild(imgTag);

tdTagTitle.textContent = book.name;
tdTagPrice.textContent = '$' + book.price + '.00';
tdTagReleaseDate.textContent = book.release_date;

let aTag = document.createElement('a');
aTag.classList.add('safe-option','change-links');
aTag.textContent = 'Change';
aTag.setAttribute('href', `index.php?page=update-book&book-
id=${book.id}`);
tdTagChange.appendChild(aTag);

let formActivate = document.createElement('form');

let inputBookIdActivate = document.createElement('input');
inputBookIdActivate.setAttribute('type','text');
inputBookIdActivate.setAttribute('name','book-id');
//inputBookIdActivate.setAttribute('value',`${book.id}`);
inputBookIdActivate.style.display = 'none';
inputBookIdActivate.value = book.id;

let inputWriterIdActivate = document.createElement('input');
inputWriterIdActivate.setAttribute('type','text');
inputWriterIdActivate.setAttribute('name','writer-id');
//inputWriterIdActivate.setAttribute('value',`${book.writer_id}`);
);
inputWriterIdActivate.style.display = 'none';
inputWriterIdActivate.value = book.writer_id;

let inputButtonActivate = document.createElement('input');
inputButtonActivate.setAttribute('type','button');
inputButtonActivate.classList.add('safe-option','activate-
book');
inputButtonActivate.value = 'Activate';

```

```

        formActivate.appendChild(inputBookIdActivate);
        formActivate.appendChild(inputWriterIdActivate);
        formActivate.appendChild(inputButtonActivate);
        tdTagActivate.appendChild(formActivate);

        trTag.appendChild(tdTagImg);
        trTag.appendChild(tdTagTitle);
        trTag.appendChild(tdTagPrice);
        trTag.appendChild(tdTagReleaseDate);
        trTag.appendChild(tdTagChange);
        trTag.appendChild(tdTagActivate);

        deletedBooksTable.appendChild(trTag);
    })

    setTableZebra();
    setEventsForWriterBooksPage();
}

function sendActivateBookFormWriter(form){
    let formData = new FormData(form);

    $.ajax({
        url: 'models/book/activate-book.php',
        type: 'POST',
        data: formData,
        contentType: false,
        processData: false,
        success: function(response) {
            populateWriterBooksTables(response);
        },
        error: function(xhr, status, errorThrown) {
            let messages = JSON.parse(xhr.responseText);
            let element = document.querySelector('.server-messages');
            element.style.display = 'block';
            displayServerMessages('server-messages',messages,'error-
message');

            console.log(messages)
            console.log(xhr);
            console.log(status, errorThrown);
        }
    });
}

function sendDeleteBookFormWriter(form){
    let formData = new FormData(form);

```

```

$.ajax({
    url: 'models/book/delete-book.php',
    type: 'POST',
    data: formData,
    contentType: false,
    processData: false,
    success: function(response) {
        populateWriterBooksTables(response);
    },
    error: function(xhr, status, errorThrown) {
        let messages = JSON.parse(xhr.responseText);
        let element = document.querySelector('.server-messages');
        element.style.display = 'block';
        displayServerMessages('server-messages',messages,'error-
message');
        console.log(messages)
        console.log(xhr);
        console.log(status, errorThrown);
    }
});

function setEventsForDeliveryTypesTable(){
    let deleteDeliveryTypes = document.querySelectorAll('.delete-
delivery-type');
    deleteDeliveryTypes.forEach(deleteDeliveryType => {
        deleteDeliveryType.addEventListener('click', () => {
            let form = deleteDeliveryType.closest('form');
            sendDeleteDeliveryTypeForm(form);
        });
    });

    let activateDeliveryTypes = document.querySelectorAll('.activate-
delivery-type');
    activateDeliveryTypes.forEach(activateDeliveryType => {
        activateDeliveryType.addEventListener('click', () => {
            let form = activateDeliveryType.closest('form');
            sendActivateDeliveryTypeForm(form);
        });
    });

    let changeLinks = document.querySelectorAll('.change-links');
    changeLinks.forEach(link=>{
        link.addEventListener('click',()=>{
            let deliveryTypeId = link.getAttribute('data-delivery-type-
id');
        });
    });
}

```

```

        let deliveryTypeName = link.getAttribute('data-delivery-
type-name');

        let inputId = document.getElementById('delivery-type-id');
        let inputName = document.getElementById('delivery-type-
name');

        inputId.setAttribute('value', deliveryTypeId);
        inputName.value = deliveryTypeName;
        document.getElementById('delivery-type-form').style.display
= 'block';
    })
})
}

function selectBookCategories(){
    let allCategories = document.querySelectorAll(".book-category-cb");
    let bookId = document.getElementById('book-id-js').value;

    $.ajax({
        url: `models/book/get-book-active-categories.php?book-
id=${bookId}`,
        type: 'GET',
        contentType: 'json',
        success: function(response) {
            console.log(response);
            for(let i=0; i<allCategories.length;i++){

                for(let j=0; j<response.length;j++){
                    if(allCategories[i].value ==
response[j].category_id){
                        allCategories[i].checked = true;
                        break;
                    }
                }
            },
            error: function(xhr, status, errorThrown) {
                let messages = JSON.parse(xhr.responseText);
                displayServerMessages('server-messages',messages,'error-
message');

                console.log(messages)
                console.log(xhr);
                console.log(status, errorThrown);
            }
        });
}

```

```

function setUpdateBookForm(){

    $(document).on('change', '#book-image-js', uploadBookImage);

    $(document).on('blur', '#book-title-title-js', ()=>{
        checkInputElementIfEmpty('book-title-title-js', 'book-title-error')
    })

    $(document).on('blur', '#page-count-input-js', ()=>{
        checkInputElementIfEmpty('page-count-input-js', 'page-count-error')
    })

    $(document).on('blur', '#price-input-js', ()=>{
        checkInputElementIfEmpty('price-input-js', 'price-error')
    })

    $(document).on('blur', '#book-description-input-js', ()=>{
        checkInputElementIfEmpty('book-description-input-js', 'book-description-error')
    })

    $(document).on('blur', '#release-date-input-js', ()=>{
        checkInputDate()
    })

    let parentCategories = document.querySelectorAll('.category-parent ul');

    parentCategories.forEach(parentCat=>{
        parentCat.style.display = 'none';
    })

    $('.category-parent').on('click', function() {
        let nestedUl = $(this).find('ul');
        nestedUl.toggle();
    });

    $('.book-category-cb').on('click', function(event) {
        event.stopPropagation();
    });

    $('.book-category-li label').on('click', function(event) {
        event.stopPropagation();
    });
}

```

```

$(document).on('click', '#update-book', sendUpdateBookData);
}

function sendUpdateBookData(){
    let errorCount = 0;

    let imageUploadInput = document.getElementById('book-image-js');
    if (imageUploadInput.files && imageUploadInput.files[0]){
        errorCount += checkBookImg();
    }

    errorCount += checkInputElementIfEmpty('book-title-title-js', 'book-
title-error')
    errorCount += checkInputElementIfEmpty('page-count-input-js', 'page-
count-error')
    errorCount += checkInputElementIfEmpty('book-description-input-
js', 'book-description-error')
    errorCount += checkInputElementIfEmpty('price-input-js', 'price-
error')
    errorCount += checkInputDate();
    errorCount += checkCategoriesCheckBoxes();

    if(errorCount){
        return;
    }

    let form = document.getElementById('update-book-form');
    let formData = new FormData(form);

    $.ajax({
        url: 'models/book/update-book.php',
        type: 'POST',
        data: formData,
        contentType: false,
        processData: false,
        success: function(response) {

            window.location.href = `index.php?page=writer-books&writer-
id=${response[0]}`;
        },
        error: function(xhr, status, errorThrown) {
            let messages = JSON.parse(xhr.responseText);
            displayServerMessages('server-messages', messages, 'error-
message');

            console.log(messages)
            console.log(xhr);
            console.log(status, errorThrown);
        }
    });
}

```

```

        }
    });
}

function sendDeleteDeliveryTypeForm(form){
    let formData = new FormData(form);

    $.ajax({
        url: 'models/delivery/delete-delivery-type.php',
        type: 'POST',
        data: formData,
        contentType: false,
        processData: false,
        success: function(response) {
            let element = document.querySelector('.server-messages');
            element.style.display = 'none';
            populateDeliveryTypesTables(response);
        },
        error: function(xhr, status, errorThrown) {
            let messages = JSON.parse(xhr.responseText);
            window.scrollTo({
                top: 0,
                behavior: 'smooth'
            });
            let element = document.querySelector('.server-messages');
            element.style.display = 'block';
            displayServerMessages('server-messages',messages,'error-
message');

            console.log(messages)
            console.log(xhr);
            console.log(status, errorThrown);
        }
    });
}

function setEventsForDeliveryTypesPage(){
    $(document).on('blur','#delivery-type-name',()=>{
        checkInputElementWithRegex(/^[A-ZŠĐĆŽ][a-zđčćž]{2,}([A-
ZŠĐĆŽ][a-zđčćž]{2,})*$/,'delivery-type-name','delivery-type-name-
error');
    })

    $(document).on('click','#cancel-button',()=>{
        document.getElementById('delivery-type-name-
error').style.display = 'none';
        document.getElementById('delivery-type-name').value = '';
    })
}

```

```

        document.getElementById('delivery-type-form').style.display =
'none';
        let inputElement = document.getElementById('delivery-type-id');
        inputElement.setAttribute('value','0');
    });

$(document).on('click','#add-delivery-type-button',()=>{
    document.getElementById('delivery-type-form').style.display =
'block';
    let inputElement = document.getElementById('delivery-type-id');
    inputElement.setAttribute('value','0');
    window.scrollTo({
        top: 0,
        behavior: 'smooth'
    });
});

$(document).on('click','#save-button',()=>{
    let form = document.getElementById('delivery-type-form');
    let errorCount = 0;
    errorCount += checkInputElementWithRegex(/^[A-ZŠĐĆŽ][a-
zšđćž]{2,}([A-ZŠĐĆŽ][a-zšđćž]{2,})*/$, 'delivery-type-
name','delivery-type-name-error');

    if(errorCount){
        return;
    }

    sendDeliveryTypeForm(form);
});
}

function populateDeliveryTypesTables(deliveryTypes){
    let activeDeliveryTypes = deliveryTypes[0].activeDeliveryTypes;
    let deletedDeliveryTypes = deliveryTypes[0].deletedDeliveryTypes;

    let activeDeliveryTypesTable = document.querySelector('#table-
active-delivery-types tbody');
    let deletedDeliveryTypesTable = document.querySelector('#table-
deleted-delivery-types tbody');

    while(activeDeliveryTypesTable.firstChild)activeDeliveryTypesTable.re
moveChild(activeDeliveryTypesTable.firstChild);
    while(deletedDeliveryTypesTable.firstChild)deletedDeliveryTypesTabl
e.removeChild(deletedDeliveryTypesTable.firstChild);

    activeDeliveryTypes.forEach(deliveryType=>{

```

```

let trTag = document.createElement('tr');

let tdTagName = document.createElement('td');
let tdTagOrderCount = document.createElement('td');
let tdTagCreatedAt = document.createElement('td');
let tdTagChange = document.createElement('td');
let tdTagDelete = document.createElement('td');

tdTagName.textContent = deliveryType.name;
tdTagOrderCount.textContent = deliveryType.order_count;
tdTagCreatedAt.textContent = deliveryType.created_at;

let aTag = document.createElement('a');
aTag.classList.add('safe-option','change-links');
aTag.setAttribute('data-delivery-type-id',deliveryType.id);
aTag.setAttribute('data-delivery-type-name',deliveryType.name);
aTag.textContent = 'Change';
aTag.setAttribute('href', `#`);
tdTagChange.appendChild(aTag);

let formDelete = document.createElement('form');

let inputDeliveryTypeIdDelete = document.createElement('input');
inputDeliveryTypeIdDelete.setAttribute('type','text');
inputDeliveryTypeIdDelete.setAttribute('name','delivery-type-
id');
inputDeliveryTypeIdDelete.style.display = 'none';
inputDeliveryTypeIdDelete.value = deliveryType.id;

let inputButtonDelete = document.createElement('input');
inputButtonDelete.setAttribute('type','button');
inputButtonDelete.classList.add('danger-option','delete-
delivery-type');
inputButtonDelete.value = 'Delete';

formDelete.appendChild(inputDeliveryTypeIdDelete);
formDelete.appendChild(inputButtonDelete);
tdTagDelete.appendChild(formDelete);

trTag.appendChild(tdTagName);
trTag.appendChild(tdTagOrderCount);
trTag.appendChild(tdTagCreatedAt);
trTag.appendChild(tdTagChange);
trTag.appendChild(tdTagDelete);

acuteDeliveryTypesTable.appendChild(trTag);
})

```

```

deletedDeliveryTypes.forEach(deliveryType=>{
    let trTag = document.createElement('tr');

    let tdTagName = document.createElement('td');
    let tdTagOrderCount = document.createElement('td');
    let tdTagDeletedAt = document.createElement('td');
    let tdTagChange = document.createElement('td');
    let tdTagActivate = document.createElement('td');

    tdTagName.textContent = deliveryType.name;
    tdTagOrderCount.textContent = deliveryType.order_count;
    tdTagDeletedAt.textContent = deliveryType.deleted_at;

    let aTag = document.createElement('a');
    aTag.setAttribute('data-delivery-type-id',deliveryType.id);
    aTag.setAttribute('data-delivery-type-name',deliveryType.name);
    aTag.classList.add('safe-option','change-links');
    aTag.textContent = 'Change';
    aTag.setAttribute('href', `#${deliveryType.id}`);
    tdTagChange.appendChild(aTag);

    let formActivate = document.createElement('form');

    let inputDeliveryTypeIdActivate =
document.createElement('input');
    inputDeliveryTypeIdActivate.setAttribute('type','text');
    inputDeliveryTypeIdActivate.setAttribute('name','delivery-type-
id');
    inputDeliveryTypeIdActivate.style.display = 'none';
    inputDeliveryTypeIdActivate.value = deliveryType.id;

    let inputButtonActivate = document.createElement('input');
    inputButtonActivate.setAttribute('type','button');
    inputButtonActivate.classList.add('safe-option','activate-
delivery-type');
    inputButtonActivate.value = 'Activate';

    formActivate.appendChild(inputDeliveryTypeIdActivate);
    formActivate.appendChild(inputButtonActivate);
    tdTagActivate.appendChild(formActivate);

    trTag.appendChild(tdTagName);
    trTag.appendChild(tdTagOrderCount);
    trTag.appendChild(tdTagDeletedAt);
    trTag.appendChild(tdTagChange);
    trTag.appendChild(tdTagActivate);

    deletedDeliveryTypesTable.appendChild(trTag);
}

```

```

        })

        setEventsForDeliveryTypesTable();
        setTableZebra();
    }

function sendActivateDeliveryTypeForm(form){
    let formData = new FormData(form);

    $.ajax({
        url: 'models/delivery/activate-delivery-type.php',
        type: 'POST',
        data: formData,
        contentType: false,
        processData: false,
        success: function(response) {
            let element = document.querySelector('.server-messages');
            element.style.display = 'none';
            populateDeliveryTypesTables(response);
        },
        error: function(xhr, status, errorThrown) {
            let messages = JSON.parse(xhr.responseText);
            window.scrollTo({
                top: 0,
                behavior: 'smooth'
            });
            let element = document.querySelector('.server-messages');
            element.style.display = 'block';
            displayServerMessages('server-messages',messages,'error-
message');

            console.log(messages)
            console.log(xhr);
            console.log(status, errorThrown);
        }
    });
}

function sendDeliveryTypeForm(form){
    let formData = new FormData(form);

    $.ajax({
        url: 'models/delivery/add-update-delivery-type.php',
        type: 'POST',
        data: formData,
        contentType: false,
        processData: false,
        success: function(response) {

```

```

        let serverMessagesContainer =
document.querySelector('.server-messages');
        serverMessagesContainer.style.display = 'none';
        document.getElementById('delivery-type-name').value = '';
        document.getElementById('delivery-type-form').style.display
= 'none';
        let inputElement = document.getElementById('delivery-type-
id');
        inputElement.setAttribute('value', '0');

        populateDeliveryTypesTables(response);
    },
error: function(xhr, status, errorThrown) {
    let messages = JSON.parse(xhr.responseText);
    window.scrollTo({
        top: 0,
        behavior: 'smooth'
    });
    let serverMessagesContainer =
document.querySelector('.server-messages');
    serverMessagesContainer.style.display = 'block';
    displayServerMessages('server-messages',messages,'error-
message');

    console.log(messages)
    console.log(xhr);
    console.log(status, errorThrown);
}
});
}

function setEventsForCategoriesPage(){
$(document).on('blur','#category-name',()=>{
    checkInputElementWithRegex(/^[A-ZŠĐĆČŽ][a-zđčćž]{2,}([A-
ZŠĐĆČŽ][a-zđčćž]{2,})*$/,'category-name','category-name-error');
});

$(document).on('click','#cancel-button',()=>{
    document.getElementById('category-name-error').style.display =
'none';
    document.getElementById('category-name').value = '';
    document.getElementById('categories-form').style.display =
'none';
    let inputElement = document.getElementById('category-id');
    inputElement.setAttribute('value','0');
});

$(document).on('click','#add-subcategory-button',()=>{

```

```

        document.getElementById('categories-form').style.display =
'block';
        let inputElement = document.getElementById('category-id');
        inputElement.setAttribute('value','0');
        document.getElementById('select-category-label').style.display
= 'block';
        document.getElementById('select-category').style.display =
'block';
        document.getElementById('category-name-error').style.display =
'none';
        let categoryType = document.getElementById('category-type');
        categoryType.setAttribute('value','subcategory');
        window.scrollTo({
            top: 0,
            behavior: 'smooth'
        });
    });

$(document).on('click', '#add-category-button', ()=>{
    document.getElementById('categories-form').style.display =
'block';
    let inputElement = document.getElementById('category-id');
    inputElement.setAttribute('value','0');
    document.getElementById('select-category-label').style.display
= 'none';
    document.getElementById('select-category').style.display =
'none';
    document.getElementById('category-name-error').style.display =
'none';

    let categoryType = document.getElementById('category-type');
    categoryType.setAttribute('value','category');
    window.scrollTo({
        top: 0,
        behavior: 'smooth'
    });
});

$(document).on('click', '#save-button', ()=>{
    let form = document.getElementById('categories-form');
    let errorCount = 0;
    errorCount += checkInputElementWithRegex(/^[A-ZŠĐĆČŽ][a-
zšđććž]{2,}([A-ZŠĐĆČŽ][a-zšđććž]{2,})*/$, 'category-name', 'category-
name-error');

    if(errorCount){
        return;
    }
})

```

```

        sendCategoryForm(form);
    });

}

function sendCategoryForm(form){
    let formData = new FormData(form);

    $.ajax({
        url: 'models/category/add-update-categories.php',
        type: 'POST',
        data: formData,
        contentType: false,
        processData: false,
        success: function(response) {
            let serverMessagesContainer =
document.querySelector('.server-messages');
            serverMessagesContainer.style.display = 'none';
            document.getElementById('category-name').value = '';
            document.getElementById('categories-form').style.display =
'none';
            let inputElement = document.getElementById('category-id');
            inputElement.setAttribute('value','0');

            populateCategoryTables(response);
        },
        error: function(xhr, status, errorThrown) {
            let messages = JSON.parse(xhr.responseText);
            window.scrollTo({
                top: 0,
                behavior: 'smooth'
            });
            let serverMessagesContainer =
document.querySelector('.server-messages');
            serverMessagesContainer.style.display = 'block';
            displayServerMessages('server-messages',messages,'error-
message');

            console.log(messages)
            console.log(xhr);
            console.log(status, errorThrown);
        }
    });
}

function populateCategoryTables(categories){
    let activeCategories = categories[0].activeCategories;
    let deletedCategories = categories[0].deletedCategories;
}

```

```

let activeCategoriesTable = document.querySelector('#table-active-categories tbody');
let ddeletedCategoriesTable = document.querySelector('#table-deleted-categories tbody');

let activeCategoriesSelect = document.querySelector('#select-category');

while(activeCategoriesSelect.firstChild)activeCategoriesSelect.removeChild(activeCategoriesSelect.firstChild);
while(activeCategoriesTable.firstChild)activeCategoriesTable.removeChild(activeCategoriesTable.firstChild);
while(ddeletedCategoriesTable.firstChild)ddeletedCategoriesTable.removeChild(ddeletedCategoriesTable.firstChild);

activeCategories.forEach(category=>{
    let trTag = document.createElement('tr');

    let tdTagName = document.createElement('td');
    let tdTagParentCategory = document.createElement('td');
    let tdTagBookCount = document.createElement('td');
    let tdTagCreatedAt = document.createElement('td');
    let tdTagChange = document.createElement('td');
    let tdTagDelete = document.createElement('td');

    tdTagName.textContent = category.name;
    tdTagBookCount.textContent = category.book_count;
    tdTagCreatedAt.textContent = category.created_at;

    tdTagParentCategory.textContent = category.parent ? category.parent : '/';

    let aTag = document.createElement('a');
    aTag.classList.add('safe-option','change-links');
    aTag.setAttribute('data-category-id',category.id);
    aTag.setAttribute('data-category-name',category.name);
    aTag.setAttribute('data-category-type',category.category_id ? 'subcategory' : 'category');
    aTag.setAttribute('data-category-parent-id',category.category_id ? category.category_id : '');
    aTag.textContent = 'Change';
    aTag.setAttribute('href',`#${category.id}`);
    tdTagChange.appendChild(aTag);

    let formDelete = document.createElement('form');

    let inputCategoryIdDelete = document.createElement('input');

```

```

    inputCategoryIdDelete.setAttribute('type', 'text');
    inputCategoryIdDelete.setAttribute('name', 'category-id');
    inputCategoryIdDelete.style.display = 'none';
    inputCategoryIdDelete.value = category.id;

    let inputButtonDelete = document.createElement('input');
    inputButtonDelete.setAttribute('type', 'button');
    inputButtonDelete.classList.add('danger-option', 'delete-
category');
    inputButtonDelete.value = 'Delete';

    formDelete.appendChild(inputCategoryIdDelete);
    formDelete.appendChild(inputButtonDelete);
    tdTagDelete.appendChild(formDelete);

    trTag.appendChild(tdTagName);
    trTag.appendChild(tdTagParentCategory);
    trTag.appendChild(tdTagBookCount);
    trTag.appendChild(tdTagCreatedAt);
    trTag.appendChild(tdTagChange);
    trTag.appendChild(tdTagDelete);

    activeCategoriesTable.appendChild(trTag);
})

deletedCategories.forEach(category=>{
    let trTag = document.createElement('tr');

    let tdTagName = document.createElement('td');
    let tdTagParentCategory = document.createElement('td');
    let tdTagBookCount = document.createElement('td');
    let tdTagDeletedAt = document.createElement('td');
    let tdTagChange = document.createElement('td');
    let tdTagActivate = document.createElement('td');

    tdTagName.textContent = category.name;
    tdTagBookCount.textContent = category.book_count;
    tdTagDeletedAt.textContent = category.deleted_at;

    tdTagParentCategory.textContent = category.parent ?
category.parent : '/';

    let aTag = document.createElement('a');
    aTag.setAttribute('data-category-id', category.id);
    aTag.setAttribute('data-category-name', category.name);
    aTag.setAttribute('data-category-type', category.category_id ?
'subcategory' : 'category');

```

```

        aTag.setAttribute('data-category-parent-
id',category.category_id ? category.category_id : '');
        aTag.classList.add('safe-option','change-links');
        aTag.textContent = 'Change';
        aTag.setAttribute('href','#');
        tdTagChange.appendChild(aTag);

        let formActivate = document.createElement('form');

        let inputCategoryIdActivate = document.createElement('input');
        inputCategoryIdActivate.setAttribute('type','text');
        inputCategoryIdActivate.setAttribute('name','category-id');
        inputCategoryIdActivate.style.display = 'none';
        inputCategoryIdActivate.value = category.id;

        let inputButtonActivate = document.createElement('input');
        inputButtonActivate.setAttribute('type','button');
        inputButtonActivate.classList.add('safe-option','activate-
category');
        inputButtonActivate.value = 'Activate';

        formActivate.appendChild(inputCategoryIdActivate);
        formActivate.appendChild(inputButtonActivate);
        tdTagActivate.appendChild(formActivate);

        trTag.appendChild(tdTagName);
        trTag.appendChild(tdTagParentCategory);
        trTag.appendChild(tdTagBookCount);
        trTag.appendChild(tdTagDeletedAt);
        trTag.appendChild(tdTagChange);
        trTag.appendChild(tdTagActivate);

        ddeletedCategoriesTable.appendChild(trTag);
    })

activeCategories.forEach(category=>{
    if(!category.category_id){
        let option = document.createElement('option');
        option.setAttribute('value',category.id);
        option.textContent = category.name;
        activeCategoriesSelect.appendChild(option);
    }
})

setEventsForCategoriesTable();
setTableZebra();
}

```

```

function setEventsForCategoriesTable(){
    let deleteCategories = document.querySelectorAll('.delete-
category');
    deleteCategories.forEach(deleteCategory => {
        deleteCategory.addEventListener('click', () => {
            let form = deleteCategory.closest('form');
            sendDeleteCategoryForm(form);
        });
    });

    let activateCategories = document.querySelectorAll('.activate-
category');
    activateCategories.forEach(activateCategory => {
        activateCategory.addEventListener('click', () => {
            let form = activateCategory.closest('form');
            sendActivateCategoryForm(form);
        });
    });

    let changeLinks = document.querySelectorAll('.change-links');
    changeLinks.forEach(link=>{
        link.addEventListener('click',()=>{
            let categoryId = link.getAttribute('data-category-id');
            let categoryrName = link.getAttribute('data-category-name');
            let categoryrType = link.getAttribute('data-category-type');
            let categoryParentId = link.getAttribute('data-category-
parent-id');

            let inputId = document.getElementById('category-id');
            let inputName = document.getElementById('category-name');
            let inputType = document.getElementById('category-type');

            inputId.setAttribute('value',categoryId);
            inputType.setAttribute('value',categoryrType);
            inputName.value = categoryrName;
            document.getElementById('categories-form').style.display =
'block';

            if(categoryrType == 'subcategory'){
                document.getElementById('select-
category').style.display = 'block';
                document.getElementById('select-category').value =
Number(categoryParentId) ;
                document.getElementById('select-category-
label').style.display = 'block';
            }else{
                document.getElementById('select-
category').style.display = 'none';
            }
        });
    });
}

```

```

        document.getElementById('select-category-
label').style.display = 'none';
    }
}
}

function setEventsForPublisherPage(){
    $(document).on('blur','#publisher-name',()=>{
        checkInputElementWithRegex(/^[A-ZŠĐĆŽ][a-zđčćž]{2,}([A-
ZŠĐĆŽ][a-zđčćž]{2,})*$/,'publisher-name','publisher-name-error');

    })

    $(document).on('click','#cancel-button',()=>{
        document.getElementById('publisher-name-error').style.display =
'none';
        document.getElementById('publisher-name').value = '';
        document.getElementById('publisher-form').style.display =
'none';
        let inputElement = document.getElementById('publisher-id');
        inputElement.setAttribute('value','0');
    });

    $(document).on('click','#add-publisher-button',()=>{
        document.getElementById('publisher-form').style.display =
'block';
        let inputElement = document.getElementById('publisher-id');
        inputElement.setAttribute('value','0');
        window.scrollTo({
            top: 0,
            behavior: 'smooth'
        });
    });

    $(document).on('click','#save-button',()=>{
        let form = document.getElementById('publisher-form');
        let errorCount = 0;
        errorCount += checkInputElementWithRegex(/^[A-ZŠĐĆŽ][a-
zđčćž]{2,}([A-ZŠĐĆŽ][a-zđčćž]{2,})*$/,'publisher-
name','publisher-name-error');

        if(errorCount){
            return;
        }

        sendPublisherForm(form);
    });
}

```

```

function sendActivateCategoryForm(form){
    let formData = new FormData(form);

    $.ajax({
        url: 'models/category/activate-category.php',
        type: 'POST',
        data: formData,
        contentType: false,
        processData: false,
        success: function(response) {
            let element = document.querySelector('.server-messages');
            element.style.display = 'none';
            populateCategoryTables(response);
        },
        error: function(xhr, status, errorThrown) {
            let messages = JSON.parse(xhr.responseText);
            window.scrollTo({
                top: 0,
                behavior: 'smooth'
            });
            let element = document.querySelector('.server-messages');
            element.style.display = 'block';
            displayServerMessages('server-messages',messages,'error-
message');
            console.log(messages)
            console.log(xhr);
            console.log(status, errorThrown);
        }
    });
}

function sendDeleteCategoryForm(form){
    let formData = new FormData(form);

    $.ajax({
        url: 'models/category/delete-category.php',
        type: 'POST',
        data: formData,
        contentType: false,
        processData: false,
        success: function(response) {
            let element = document.querySelector('.server-messages');
            element.style.display = 'none';
            populateCategoryTables(response);
        },
        error: function(xhr, status, errorThrown) {
            let messages = JSON.parse(xhr.responseText);

```

```

        window.scrollTo({
            top: 0,
            behavior: 'smooth'
        });
        let element = document.querySelector('.server-messages');
        element.style.display = 'block';
        displayServerMessages('server-messages', messages, 'error-
message');

        console.log(messages)
        console.log(xhr);
        console.log(status, errorThrown);
    }
});

function setEventsForPublisherTable(){
    let deletePublishers = document.querySelectorAll('.delete-
publisher');
    deletePublishers.forEach(deletePublisher => {
        deletePublisher.addEventListener('click', () => {
            let form = deletePublisher.closest('form');
            sendDeletePublisherForm(form);
        });
    });

    let activatePublishers = document.querySelectorAll('.activate-
publisher');
    activatePublishers.forEach(activatePublisher => {
        activatePublisher.addEventListener('click', () => {
            let form = activatePublisher.closest('form');
            sendActivatePublisherForm(form);
        });
    });

    let changeLinks = document.querySelectorAll('.change-links');
    changeLinks.forEach(link=>{
        link.addEventListener('click',()=>{
            let publisherId = link.getAttribute('data-publisher-id');
            let publisherName = link.getAttribute('data-publisher-
name');

            let inputId = document.getElementById('publisher-id');
            let inputName = document.getElementById('publisher-name');

            inputId.setAttribute('value',publisherId);
            inputName.value = publisherName;
        });
    });
}

```

```

        document.getElementById('publisher-form').style.display =
'block';
    })
})
}

function sendActivatePublisherForm(form){
    let formData = new FormData(form);

    $.ajax({
        url: 'models/publisher/activate-publisher.php',
        type: 'POST',
        data: formData,
        contentType: false,
        processData: false,
        success: function(response) {
            let element = document.querySelector('.server-messages');
            element.style.display = 'none';
            populatePublisherTables(response);
        },
        error: function(xhr, status, errorThrown) {
            let messages = JSON.parse(xhr.responseText);
            window.scrollTo({
                top: 0,
                behavior: 'smooth'
            });
            let element = document.querySelector('.server-messages');
            element.style.display = 'block';
            displayServerMessages('server-messages',messages,'error-
message');

            console.log(messages)
            console.log(xhr);
            console.log(status, errorThrown);
        }
    });
}

function populatePublisherTables(publishers){
    let activePublishers = publishers[0].activePublishers;
    let deletedPublishers = publishers[0].deletedPublishers;

    let activePublishersTable = document.querySelector('#table-active-
publishers tbody');
    let deletedPublishersTable = document.querySelector('#table-
deleted-publishers tbody');

```

```

        while(acitePublishersTable.firstChild)acitePublishersTable.removeChild(acitePublishersTable.firstChild);
        while(deletedPublishersTable.firstChild)deletedPublishersTable.removeChild(deletedPublishersTable.firstChild);

        activePublishers.forEach(publisher=>{
            let trTag = document.createElement('tr');

            let tdTagName = document.createElement('td');
            let tdTagBookCount = document.createElement('td');
            let tdTagCreatedAt = document.createElement('td');
            let tdTagChange = document.createElement('td');
            let tdTagDelete = document.createElement('td');

            tdTagName.textContent = publisher.name;
            tdTagBookCount.textContent = publisher.book_count;
            tdTagCreatedAt.textContent = publisher.created_at;

            let aTag = document.createElement('a');
            aTag.classList.add('safe-option','change-links');
            aTag.setAttribute('data-publisher-id',publisher.id);
            aTag.setAttribute('data-publisher-name',publisher.name);
            aTag.textContent = 'Change';
            aTag.setAttribute('href',`#`);
            tdTagChange.appendChild(aTag);

            let formDelete = document.createElement('form');

            let inputPublisherIdDelete = document.createElement('input');
            inputPublisherIdDelete.setAttribute('type','text');
            inputPublisherIdDelete.setAttribute('name','publisher-id');
            inputPublisherIdDelete.style.display = 'none';
            inputPublisherIdDelete.value = publisher.id;

            let inputButtonDelete = document.createElement('input');
            inputButtonDelete.setAttribute('type','button');
            inputButtonDelete.classList.add('danger-option','delete-publisher');
            inputButtonDelete.value = 'Delete';

            formDelete.appendChild(inputPublisherIdDelete);
            formDelete.appendChild(inputButtonDelete);
            tdTagDelete.appendChild(formDelete);

            trTag.appendChild(tdTagName);
            trTag.appendChild(tdTagBookCount);
            trTag.appendChild(tdTagCreatedAt);
            trTag.appendChild(tdTagChange);
        });
    }
}

```

```

        trTag.appendChild(tdTagDelete);

        acutePublishersTable.appendChild(trTag);
    })

    deletedPublishers.forEach(publisher=>{
        let trTag = document.createElement('tr');

        let tdTagName = document.createElement('td');
        let tdTagBookCount = document.createElement('td');
        let tdTagDeletedAt = document.createElement('td');
        let tdTagChange = document.createElement('td');
        let tdTagActivate = document.createElement('td');

        tdTagName.textContent = publisher.name;
        tdTagBookCount.textContent = publisher.book_count;
        tdTagDeletedAt.textContent = publisher.deleted_at;

        let aTag = document.createElement('a');
        aTag.setAttribute('data-publisher-id',publisher.id);
        aTag.setAttribute('data-publisher-name',publisher.name);
        aTag.classList.add('safe-option','change-links');
        aTag.textContent = 'Change';
        aTag.setAttribute('href',`#`);
        tdTagChange.appendChild(aTag);

        let formActivate = document.createElement('form');

        let inputPublisherIdActivate = document.createElement('input');
        inputPublisherIdActivate.setAttribute('type','text');
        inputPublisherIdActivate.setAttribute('name','publisher-id');
        inputPublisherIdActivate.style.display = 'none';
        inputPublisherIdActivate.value = publisher.id;

        let inputButtonActivate = document.createElement('input');
        inputButtonActivate.setAttribute('type','button');
        inputButtonActivate.classList.add('safe-option','activate-
publisher');
        inputButtonActivate.value = 'Activate';

        formActivate.appendChild(inputPublisherIdActivate);
        formActivate.appendChild(inputButtonActivate);
        tdTagActivate.appendChild(formActivate);

        trTag.appendChild(tdTagName);
        trTag.appendChild(tdTagBookCount);
        trTag.appendChild(tdTagDeletedAt);
        trTag.appendChild(tdTagChange);
    })
}

```

```

        trTag.appendChild(tdTagActivate);

        deletedPublishersTable.appendChild(trTag);
    })

    setEventsForPublisherTable();
    setTableZebra();
}

function sendDeletePublisherForm(form){
    let formData = new FormData(form);

    $.ajax({
        url: 'models/publisher/delete-publisher.php',
        type: 'POST',
        data: formData,
        contentType: false,
        processData: false,
        success: function(response) {
            let element = document.querySelector('.server-messages');
            element.style.display = 'none';
            populatePublisherTables(response);
        },
        error: function(xhr, status, errorThrown) {
            let messages = JSON.parse(xhr.responseText);
            window.scrollTo({
                top: 0,
                behavior: 'smooth'
            });
            let element = document.querySelector('.server-messages');
            element.style.display = 'block';
            displayServerMessages('server-messages',messages,'error-
message');

            console.log(messages)
            console.log(xhr);
            console.log(status, errorThrown);
        }
    });
}

function sendPublisherForm(form){
    let formData = new FormData(form);

    $.ajax({
        url: 'models/publisher/add-update-publisher.php',
        type: 'POST',
        data: formData,

```

```

        contentType: false,
        processData: false,
        success: function(response) {
            let serverMessagesContainer =
document.querySelector('.server-messages');
            serverMessagesContainer.style.display = 'none';
            document.getElementById('publisher-name').value = '';
            document.getElementById('publisher-form').style.display =
'nave';
            let inputElement = document.getElementById('publisher-id');
            inputElement.setAttribute('value','0');
            populatePublisherTables(response);
        },
        error: function(xhr, status, errorThrown) {
            let messages = JSON.parse(xhr.responseText);
            window.scrollTo({
                top: 0,
                behavior: 'smooth'
            });
            let serverMessagesContainer =
document.querySelector('.server-messages');
            serverMessagesContainer.style.display = 'block';
            displayServerMessages('server-messages',messages,'error-
message');
            console.log(messages)
            console.log(xhr);
            console.log(status, errorThrown);
        }
    });
}

function setEventsForUserTable(){
    let deleteUsers = document.querySelectorAll('.delete-user');
    deleteUsers.forEach(deleteUser => {
        deleteUser.addEventListener('click', () => {
            let form = deleteUser.closest('form');
            sendDeleteUserForm(form);
        });
    });

    let activateUsers = document.querySelectorAll('.activate-user');
    activateUsers.forEach(activateUser => {
        activateUser.addEventListener('click', () => {
            let form = activateUser.closest('form');
            sendActivateUserForm(form);
        });
    });
}

```

```

let banUsers = document.querySelectorAll('.ban-user');
banUsers.forEach(banUser => {
    banUser.addEventListener('click', () => {
        let form = banUser.closest('form');
        sendBanUserForm(form);
    });
});

let unbanUsers = document.querySelectorAll('.unban-user');
unbanUsers.forEach(unbanUser => {
    unbanUser.addEventListener('click', () => {
        let form = unbanUser.closest('form');
        sendUnbanUserForm(form);
    });
});
}

function populateUserTables(users){
    let activeUsers = users[0].activeUsers;
    let deletedUsers = users[0].deletedUsers;
    let bannedUsers = users[0].bannedUsers;

    let activeUsersTable = document.querySelector('#table-active-users tbody');
    let deletedUsersTable = document.querySelector('#table-deleted-users tbody');
    let bannedUsersTable = document.querySelector('#table-banned-users tbody');

    while(activeUsersTable.firstChild)activeUsersTable.removeChild(activeUsersTable.firstChild);
    while(deletedUsersTable.firstChild)deletedUsersTable.removeChild(deletedUsersTable.firstChild);
    while(bannedUsersTable.firstChild)bannedUsersTable.removeChild(bannedUsersTable.firstChild);

    activeUsers.forEach(user=>{
        let trTag = document.createElement('tr');

        let tdTagUsername = document.createElement('td');
        let tdTagEmail = document.createElement('td');
        let tdTagRole = document.createElement('td');
        let tdTagCreatedAt = document.createElement('td');
        let tdTagChange = document.createElement('td');
        let tdTagBan = document.createElement('td');
        let tdTagDelete = document.createElement('td');

        tdTagUsername.textContent = user.username;

```

```

tdTagEmail.textContent = user.email;
tdTagRole.textContent = user.name;
tdTagCreatedAt.textContent = user.created_at;

let aTag = document.createElement('a');
aTag.classList.add('safe-option');
aTag.textContent = 'Change';
aTag.setAttribute('href', `index.php?page=user-profile&selected-
user=${user.id}`);
tdTagChange.appendChild(aTag);

let formBan = document.createElement('form');

let inputUserId = document.createElement('input');
inputUserId.setAttribute('type', 'text');
inputUserId.setAttribute('name', 'userId');
inputUserId.style.display = 'none';
inputUserId.value = user.id;

let inputButtonBan = document.createElement('input');
inputButtonBan.setAttribute('type', 'button');
inputButtonBan.classList.add('danger-option', 'ban-user');
inputButtonBan.value = 'Ban';

formBan.appendChild(inputUserId);
formBan.appendChild(inputButtonBan);
tdTagBan.appendChild(formBan);

let formBDelete = document.createElement('form');

let inputUserIdDelete = document.createElement('input');
inputUserIdDelete.setAttribute('type', 'text');
inputUserIdDelete.setAttribute('name', 'userId');
inputUserIdDelete.style.display = 'none';
inputUserIdDelete.value = user.id;

let inputButtonDelete = document.createElement('input');
inputButtonDelete.setAttribute('type', 'button');
inputButtonDelete.classList.add('danger-option', 'delete-user');
inputButtonDelete.value = 'Delete';

formBDelete.appendChild(inputUserIdDelete);
formBDelete.appendChild(inputButtonDelete);
tdTagDelete.appendChild(formBDelete);

trTag.appendChild(tdTagUsername);
trTag.appendChild(tdTagEmail);
trTag.appendChild(tdTagRole);

```

```

        trTag.appendChild(tdTagCreatedAt);
        trTag.appendChild(tdTagChange);
        trTag.appendChild(tdTagBan);
        trTag.appendChild(tdTagDelete);

        acuteUsersTable.appendChild(trTag);
    })

deletedUsers.forEach(user=>{
    let trTag = document.createElement('tr');

    let tdTagUsername = document.createElement('td');
    let tdTagEmail = document.createElement('td');
    let tdTagRole = document.createElement('td');
    let tdTagCreatedAt = document.createElement('td');
    let tdTagDeletedAt = document.createElement('td');
    let tdTagChange = document.createElement('td');
    let tdTagActivate = document.createElement('td');

    tdTagUsername.textContent = user.username;
    tdTagEmail.textContent = user.email;
    tdTagRole.textContent = user.name;
    tdTagCreatedAt.textContent = user.created_at;
    tdTagDeletedAt.textContent = user.deleted_at;

    let aTag = document.createElement('a');
    aTag.classList.add('safe-option');
    aTag.textContent = 'Change';
    aTag.setAttribute('href', `index.php?page=user-profile&selected-
user=${user.id}`);
    tdTagChange.appendChild(aTag);

    let formActivate = document.createElement('form');

    let inputUserId = document.createElement('input');
    inputUserId.setAttribute('type', 'text');
    inputUserId.setAttribute('name', 'userId');
    inputUserId.style.display = 'none';
    inputUserId.value = user.id;

    let inputButtonActivate = document.createElement('input');
    inputButtonActivate.setAttribute('type', 'button');
    inputButtonActivate.classList.add('safe-option', 'activate-
user');
    inputButtonActivate.value = 'Activate';

    formActivate.appendChild(inputUserId);
    formActivate.appendChild(inputButtonActivate);
})

```

```

        tdTagActivate.appendChild(formActivate);

        trTag.appendChild(tdTagUsername);
        trTag.appendChild(tdTagEmail);
        trTag.appendChild(tdTagRole);
        trTag.appendChild(tdTagCreatedAt);
        trTag.appendChild(tdTagDeletedAt);
        trTag.appendChild(tdTagChange);
        trTag.appendChild(tdTagActivate);

        deletedUsersTable.appendChild(trTag);
    })

bannedUsers.forEach(user=>{
    let trTag = document.createElement('tr');

    let tdTagUsername = document.createElement('td');
    let tdTagEmail = document.createElement('td');
    let tdTagRole = document.createElement('td');
    let tdTagCreatedAt = document.createElement('td');
    let tdTagBannedAt = document.createElement('td');
    let tdTagChange = document.createElement('td');
    let tdTagAUnban = document.createElement('td');

    tdTagUsername.textContent = user.username;
    tdTagEmail.textContent = user.email;
    tdTagRole.textContent = user.name;
    tdTagCreatedAt.textContent = user.created_at;
    tdTagBannedAt.textContent = user.banned_at;

    let aTag = document.createElement('a');
    aTag.classList.add('safe-option');
    aTag.textContent = 'Change';
    aTag.setAttribute('href', `index.php?page=user-profile&selected-
user=${user.id}`);
    tdTagChange.appendChild(aTag);

    let formUnban = document.createElement('form');

    let inputUserId = document.createElement('input');
    inputUserId.setAttribute('type', 'text');
    inputUserId.setAttribute('name', 'userId');
    inputUserId.style.display = 'none';
    inputUserId.value = user.id;

    let inputButtonUnban = document.createElement('input');
    inputButtonUnban.setAttribute('type', 'button');
    inputButtonUnban.classList.add('safe-option', 'unban-user');

```

```

        inputButtonUnban.value = 'Unban';

        formUnban.appendChild(inputUserId);
        formUnban.appendChild(inputButtonUnban);
        tdTagAUnban.appendChild(formUnban);

        trTag.appendChild(tdTagUsername);
        trTag.appendChild(tdTagEmail);
        trTag.appendChild(tdTagRole);
        trTag.appendChild(tdTagCreatedAt);
        trTag.appendChild(tdTagBannedAt);
        trTag.appendChild(tdTagChange);
        trTag.appendChild(tdTagAUnban);

        bannedUsersTable.appendChild(trTag);
    })

    setEventsForUserTable();
    setTableZebra();
}

function sendUnbanUserForm(form){
    let formData = new FormData(form);

    $.ajax({
        url: 'models/user/unban-user.php',
        type: 'POST',
        data: formData,
        contentType: false,
        processData: false,
        success: function(response) {
            let element = document.querySelector('.server-messages');
            element.style.display = 'none';
            populateUserTables(response);
        },
        error: function(xhr, status, errorThrown) {
            let messages = JSON.parse(xhr.responseText);
            window.scrollTo({
                top: 0,
                behavior: 'smooth'
            });
            let element = document.querySelector('.server-messages');
            element.style.display = 'block';
            displayServerMessages('server-messages',messages,'error-
message');

            console.log(messages)
            console.log(xhr);
        }
    })
}

```

```

        console.log(status, errorThrown);
    }
});
}

function sendBanUserForm(form){
    let formData = new FormData(form);

    $.ajax({
        url: 'models/user/ban-user.php',
        type: 'POST',
        data: formData,
        contentType: false,
        processData: false,
        success: function(response) {
            let element = document.querySelector('.server-messages');
            element.style.display = 'none';
            populateUserTables(response);
        },
        error: function(xhr, status, errorThrown) {
            let messages = JSON.parse(xhr.responseText);
            let element = document.querySelector('.server-messages');
            element.style.display = 'block';
            window.scrollTo({
                top: 0,
                behavior: 'smooth'
            });
            displayServerMessages('server-messages',messages,'error-
message');
            console.log(messages)
            console.log(xhr);
            console.log(status, errorThrown);
        }
    });
}

function sendActivateUserForm(form){
    let formData = new FormData(form);

    $.ajax({
        url: 'models/user/activate-user.php',
        type: 'POST',
        data: formData,
        contentType: false,
        processData: false,
        success: function(response) {
            let element = document.querySelector('.server-messages');

```

```

        element.style.display = 'none';
        populateUserTables(response);
    },
    error: function(xhr, status, errorThrown) {
        let messages = JSON.parse(xhr.responseText);
        let element = document.querySelector('.server-messages');
        element.style.display = 'block';
        window.scrollTo({
            top: 0,
            behavior: 'smooth'
        });
        displayServerMessages('server-messages',messages,'error-
message');

        console.log(messages)
        console.log(xhr);
        console.log(status, errorThrown);
    }
});

function sendDeleteUserForm(form){
    let formData = new FormData(form);

    $.ajax({
        url: 'models/user/delete-user.php',
        type: 'POST',
        data: formData,
        contentType: false,
        processData: false,
        success: function(response) {
            let element = document.querySelector('.server-messages');
            element.style.display = 'none';
            populateUserTables(response);
        },
        error: function(xhr, status, errorThrown) {
            let messages = JSON.parse(xhr.responseText);
            let element = document.querySelector('.server-messages');
            element.style.display = 'block';
            window.scrollTo({
                top: 0,
                behavior: 'smooth'
            });
            displayServerMessages('server-messages',messages,'error-
message');

            console.log(messages)
            console.log(xhr);
        }
    });
}

```

```

        console.log(status, errorThrown);
    }
});

function setTableZebra(){
    $("tbody tr:odd").css("background-color","var(--table-tr-bg)");
}

function setUpdateUserForm(){
    $(document).on('change','#user-avatar-js',uploadAvatar);
    $(document).on('blur','#biography-input-js',checkBiography);

    $(document).on('blur','#first-name-input-js',()=>{
        checkInputElementWithRegex(/^[A-ZŠĐĆŽ][a-zđčćž]{2,}([A-ZŠĐĆŽ][a-zđčćž]{2,})*/$, 'first-name-input-js','first-name-error');
    });
    $(document).on('blur','#last-name-input-js',()=>{
        checkInputElementWithRegex(/^[A-ZŠĐĆŽ][a-zđčćž]{2,}([A-ZŠĐĆŽ][a-zđčćž]{2,})*/$, 'last-name-input-js','last-name-error');
    });
    $(document).on('blur','#username-input-js',()=>{
        checkInputElementWithRegex(/^[_a-zA-Z0-9.šđčćčŠĐŽ\(\)\/\-\_]{5,}$/, 'username-input-js','username-error');
    });
    $(document).on('blur','#email-input-js',()=>{
        checkInputElementWithRegex(/^[_a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$/, 'email-input-js','email-error');
    });
    $(document).on('blur','#phone-input-js',()=>{
        checkInputElementWithRegex(/^\\d{5,15}$/, 'phone-input-js','phone-error');
    });
    $(document).on('blur','#address-line-input-js',()=>{
        checkInputElementWithRegex(/^[_a-zA-Zđčćž]+[_a-zA-Zđčćž0-9\\s\\.\\-\\#\\\\/\\\,]+$/,'address-line-input-js','address-line-error');
    });
    $(document).on('blur','#number-input-js',()=>{
        checkInputElementWithRegex(/^\\d+[a-zA-Z]?$/,'number-input-js','number-error');
    });
    $(document).on('blur','#city-input-js',()=>{
        checkInputElementWithRegex(/^([A-ZŠĐĆŽ][a-zđčćž]{2,}\\s?)+$/,'city-input-js','city-error');
    });
    $(document).on('blur','#state-input-js',()=>{
        checkInputElementWithRegex(/^([A-ZŠĐĆŽ][a-zđčćž]{2,}\\s?)+$/,'state-input-js','state-error');
    });
}

```

```

    });
    $(document).on('blur', '#country-input-js', ()=>{
        checkInputElementWithRegex(/^(A-ZŠĐĆČŽ][a-
zšđććž]{2,}\s?)+$/,'country-input-js','country-error');
    });
    $(document).on('blur', '#zip-code-input-js', ()=>{
        checkInputElementWithRegex(/^\d{5,15}$/, 'zip-code-input-
js','zip-code-error');
    });

    $(document).on('click', '#save-account-button', saveUpdatedUserData);
}

function saveUpdatedUserData(){
    let errorCount = 0;

    let adresLine = document.getElementById('address-line-input-
js').value;
    let adresNumber = document.getElementById('number-input-js').value;
    let city = document.getElementById('city-input-js').value;
    let state = document.getElementById('state-input-js').value;
    let zipCode = document.getElementById('zip-code-input-js').value;
    let country = document.getElementById('country-input-js').value;
    let role = document.getElementById('role-input').value

    errorCount += checkAvatar();
    errorCount += checkInputElementWithRegex(/^[A-ZŠĐĆČŽ][a-
zšđććž]{2,}([A-ZŠĐĆČŽ][a-zšđććž]{2,})*$/,'first-name-input-js','first-
name-error');
    errorCount += checkInputElementWithRegex(/^[A-ZŠĐĆČŽ][a-
zšđććž]{2,}([A-ZŠĐĆČŽ][a-zšđććž]{2,})*$/,'last-name-input-js','last-
name-error');
    errorCount += checkInputElementWithRegex(/^a-zA-Z0-
9.šđććšđž(\()\|[_]{5,}$/, 'username-input-js','username-error');
    errorCount += checkInputElementWithRegex(/^a-zA-Z0-9._%+-]+@[a-zA-
Z0-9.-]+\.[a-zA-Z]{2,}$/, 'email-input-js','email-error');
    errorCount += checkInputElementWithRegex(/^\\d{5,15}$/, 'phone-input-
js','phone-error');
    if(role == 3){
        errorCount += checkBiography();
    }

    if(adresLine.length || adresNumber.length || city.length ||
state.length || zipCode.length || country.length){
        errorCount += checkInputElementWithRegex(/^a-zA-Z0-
9\s\.\-\#\|\|\,]+$/,'address-line-input-js','address-line-error');
        errorCount += checkInputElementWithRegex(/^\\d+[a-zA-
Z]?\$/,'number-input-js','number-error');
    }
}

```

```

        errorCount += checkInputElementWithRegex(/^(A-ZŠĐŽĆČ][a-
zšđžćč]{2,}\s?)+$/,'city-input-js','city-error');
        errorCount += checkInputElementWithRegex(/^(A-ZŠĐĆĆŽ][a-
zšđčćž]{2,}\s?)+$/,'state-input-js','state-error');
        errorCount += checkInputElementWithRegex(/^(A-ZŠĐĆĆŽ][a-
zšđčćž]{2,}\s?)+$/,'country-input-js','country-error');
        errorCount += checkInputElementWithRegex(/^\\d{5,15}$/, 'zip-
code-input-js','zip-code-error');
    }

    if(errorCount){
        return;
    }

    let form = document.getElementById('update-user-form');
    let formData = new FormData(form);

    $.ajax({
        url: 'models/user/update-user-info.php',
        type: 'POST',
        data: formData,
        contentType: false,
        processData: false,
        success: function(response) {
            location.reload();
        },
        error: function(xhr, status, errorThrown) {
            let messages = JSON.parse(xhr.responseText);
            displayServerMessages('error-server-
messages',messages,'error-message');

            console.log(messages)
            console.log(xhr);
            console.log(status, errorThrown);
        }
    });
}

function setUserAccount(){
    $(document).on('click','#update-account-button',()=>{
        document.getElementById('user-data').style.display = 'none';
        document.getElementById('update-user-form').style.display =
'block';

        cleareAllServerMessages('success-server-messages');

        window.scrollTo({
            top: 0,

```

```

        behavior: 'smooth'
    });
});

$(document).on('click', '#cancel-account-button', ()=>{
    setUserData();

    document.getElementById('user-data').style.display = 'block';
    document.getElementById('update-user-form').style.display =
'none';

    cleareAllServerMessages('error-server-messages');
    clearAllInputErrors();
});

$(document).on('click', '#delete-account-button', deleteUser)
}

function deleteUser(){
    if(window.confirm("This account will be permanently deleted.")){
        let userId = document.getElementById('user-id').value;
        let data = {
            userId : userId
        }
        $.ajax({
            url: 'models/user/delete-user.php',
            type: 'POST',
            dataType:'json',
            data: data,
            success: function(response) {
                window.location.href = 'index.php?page=home';
            },
            error: function(xhr, status, errorThrown) {
                let messages = JSON.parse(xhr.responseText);
                displayServerMessages('server-
messages',messages,'error-message');

                console.log(messages)
                console.log(xhr);
                console.log(status, errorThrown);
            }
        });
    }
}

function setUserData(){

    document.getElementById('user-avatar-js').value = "";
}

```

```
let avatarImg = document.getElementById('user-avatar-img');

let cookie = document.cookie.split(';');

for(let i=0; i<cookie.length;i++){

    let name = cookie[i].trim().split('=')[0];
    let value = decodeURIComponent(cookie[i].trim().split('=')[1]);

    switch(name){
        case 'avatarSrc':
            avatarImg.setAttribute('src','../assets/images/avatars/${value}`);
            break;
        case 'firstName':
            document.getElementById('first-name-input-js').value = value;
            document.querySelector('p.first-name-input').textContent = value;
            break;
        case 'lastName':
            document.getElementById('last-name-input-js').value = value;
            document.querySelector('p.last-name-input').textContent = value;
            break;
        case 'username':
            document.getElementById('username-input-js').value = value;
            document.querySelector('p.username-input').textContent = value;
            break;
        case 'email':
            document.getElementById('email-input-js').value = value;
            document.querySelector('p.email-input').textContent = value;
            break;
        case 'phone':
            document.getElementById('phone-input-js').value = value;
            document.querySelector('p.phone-input').textContent = value;
            break;
        case 'biographyText':
            document.getElementById('biography-input-js').value = value;
            document.querySelector('p.biography-input').textContent = value;
            break;
    }
}
```

```

        case 'addressLine':
            document.getElementById('address-line-input-js').value =
value == '/' ? '' : value;
            document.querySelector('p.address-line-
input').textContent = value;
            break;
        case 'addressNumber':
            document.getElementById('number-input-js').value =
value == '/' ? '' : value;
            document.querySelector('p.number-input').textContent =
value;
            break;
        case 'city':
            document.getElementById('city-input-js').value = value
== '/' ? '' : value;
            document.querySelector('p.city-input').textContent =
value;
            break;
        case 'state':
            document.getElementById('state-input-js').value = value
== '/' ? '' : value;
            document.querySelector('p.state-input').textContent =
value;
            break;
        case 'zipCode':
            document.getElementById('zip-code-input-js').value =
value == '/' ? '' : value;
            document.querySelector('p.zip-code-input').textContent =
value;
            break;
        case 'country':
            document.getElementById('country-input-js').value =
value == '/' ? '' : value;
            document.querySelector('p.country-input').textContent =
value;
            break;
    }

}

}

function setLoginForm(){
$(document).on('blur','#username-input-js',()=>{
    checkInputElementWithRegex(/^[a-zA-Z0-9.šđččĆĆŠĐŽ\(\)\/\-\_]{5,}$/, '#username-input-js', 'username-error');
});
$(document).on('blur','#password-input-js',()=>{

```

```

        checkInputElementWithRegex(/^(?=.*[a-zđčćž])(?=.*[A-
ZČĆŽĐ])(?=.*\d)(?=.*[_()/-])[A-ZĐĆŽČa-zđčćž\d._()/-_
]{5,}$/, 'password-input-js', 'password-error');
    });

    $(document).on('click', '#log-in-button', sendLoginData);
}

function sendLoginData(){
    let errorCount = 0;

    errorCount += checkInputElementWithRegex(/^[a-zA-Z0-
9.šđžćčĆŠĐŽ\(\)\/_]{5,}$/, 'username-input-js', 'username-error');

    errorCount += checkInputElementWithRegex(/^(?=.*[a-zđčćž])(?=.*[A-
ZČĆŽĐ])(?=.*\d)(?=.*[_()/-])[A-ZĐĆŽČa-zđčćž\d._()/-_
]{5,}$/, 'password-input-js', 'password-error');

    if(errorCount){
        return;
    }

    let form = document.getElementById('login-form');
    let formData = new FormData(form);

    $.ajax({
        url: 'models/login/login.php',
        type: 'POST',
        data: formData,
        contentType: false,
        processData: false,
        success: function(response) {
            // console.log(response);
            window.location.href = 'index.php?page=home';
        },
        error: function(xhr, status, errorThrown) {
            let messages = JSON.parse(xhr.responseText);
            displayServerMessages('server-messages', messages, 'error-
message');

            console.log(messages)
            console.log(xhr);
            console.log(status, errorThrown);
        }
    });
}

function checkBookImg(){
    let imageUploadInput = document.getElementById('book-image-js');

```

```

let errorMessage = document.getElementById('book-image-error');

if (imageUploadInput.files && imageUploadInput.files[0]){

    let file = imageUploadInput.files[0];

    let validExtensions = ["jpg", "jpeg", "png"];
    let fileExtension = file.name.split(".").pop().toLowerCase();

    if (!validExtensions.includes(fileExtension)){
        errorMessage.style.display = "block";
        return 1;
    }

    let fileSizeInKB = file.size / 1024;
    if (fileSizeInKB > 700) {
        errorMessage.style.display = "block";
        return 1;
    }

    let img = new Image();
    img.src = URL.createObjectURL(file);

    img.onload = function(){
        if(img.height/img.width < 200/170){
            errorMessage.style.display = "block";
            return 1;
        }
    }
} else{
    errorMessage.style.display = "block";
    return 1;
}

errorMessage.style.display = "none";
return 0;
}

function uploadBookImage(){
    let bookImgContainer = document.querySelector('.book-image-
container-form');
    let imageUploadInput = document.getElementById('book-image-js');

    if (imageUploadInput.files && imageUploadInput.files[0]
&& !checkBookImg()){
        while(bookImgContainer.firstChild)bookImgContainer.removeChild(
bookImgContainer.firstChild);
    }
}

```

```

let file = imageUploadInput.files[0];
let fileURL = URL.createObjectURL(file);

let imgTag = document.createElement('img');
imgTag.src = fileURL;

bookImgContainer.appendChild(imgTag)
}else{
    while(bookImgContainer.firstChild)bookImgContainer.removeChild(
bookImgContainer.firstChild);
}
}

function checkInputElementIfEmpty(idElement,idError){
    let element = document.getElementById(idElement);
    let error = document.getElementById(idError);

    if(element.value == ""){
        error.style.display = "block";
        return 1;
    }

    error.style.display = "none";
    return 0;
}

function checkInputDate(){
    let today = new Date();

    let input = document.getElementById('release-date-input-js').value;

    if (!input) {
        document.getElementById('release-date-error').style.display =
'block'
        return 1;
    }

    let inputDate = new Date(input);

    if (inputDate > today) {

        document.getElementById('release-date-error').style.display =
'block'
        return 1;
    }

    document.getElementById('release-date-error').style.display =
'none'
}

```

```

        return 0;
    }

function checkCategoriesCheckBoxes(){
    let checkboxes = document.querySelectorAll('.book-category-cb');

    let isChecked = false;

    checkboxes.forEach((checkbox) => {
        if (checkbox.checked) {
            isChecked = true;
        }
    });

    if(isChecked){
        document.getElementById('book-category-error').style.display =
'none';
        return 0;
    }

    document.getElementById('book-category-error').style.display =
'block';
    return 1;
}

function setInsertBookForm(){

    $(document).on('change', '#book-image-js',uploadBookImage);

    $(document).on('blur', '#book-title-title-js',()=>{
        checkInputElementIfEmpty('book-title-title-js','book-title-
error')
    })

    $(document).on('blur', '#page-count-input-js',()=>{
        checkInputElementIfEmpty('page-count-input-js','page-count-
error')
    })

    $(document).on('blur', '#price-input-js',()=>{
        checkInputElementIfEmpty('price-input-js','price-error')
    })

    $(document).on('blur', '#book-description-input-js',()=>{
        checkInputElementIfEmpty('book-description-input-js','book-
description-error')
    })
}

```

```

$(document).on('blur', '#release-date-input-js', ()=>{
    checkInputDate()
})

let parentCategories = document.querySelectorAll('.category-parent ul');

parentCategories.forEach(parentCat=>{
    parentCat.style.display = 'none';
})

$('.category-parent').on('click', function() {
    var nestedUl = $(this).find('ul');
    nestedUl.toggle();
});

$('.book-category-cb').on('click', function(event) {
    event.stopPropagation();
});

$('.book-category-li label').on('click', function(event) {
    event.stopPropagation();
});

$(document).on('click', '#insert-new-book', sendInsertBookData);
}

function clearAllInputBookValues(){
    let bookImgContainer = document.querySelector('.book-image-container-form');
    while(bookImgContainer.firstChild)bookImgContainer.removeChild(bookImgContainer.firstChild);

    document.getElementById('book-image-js').value = "";

    document.getElementById('book-title-title-js').value = "";
    document.getElementById('book-description-input-js').value = "";
    document.getElementById('page-count-input-js').value = "";
    document.getElementById('price-input-js').value = "";
    document.getElementById('release-date-input-js').value = "";
    document.getElementById('publisher-input-js').value = 1;

    let checkboxes = document.querySelectorAll('.book-category-cb');

    checkboxes.forEach((checkbox) => {
        checkbox.checked = false;
    });
}

```

```

}

function clearAllInputBookErrors(){
    document.getElementById('book-image-error').style.display = "none";
    document.getElementById('book-title-error').style.display = "none";
    document.getElementById('page-count-error').style.display = "none";
    document.getElementById('price-error').style.display = "none";
    document.getElementById('release-date-error').style.display =
"none";
    document.getElementById('publisher-error').style.display = "none";
    document.getElementById('book-category-error').style.display =
"none";
    document.getElementById('book-description-error').style.display =
"none";

}

function sendInsertBookData(){
    let errorCount = 0;

    errorCount += checkBookImg();
    errorCount += checkInputElementIfEmpty('book-title-title-js', 'book-
title-error')
    errorCount += checkInputElementIfEmpty('page-count-input-js', 'page-
count-error')
    errorCount += checkInputElementIfEmpty('book-description-input-
js', 'book-description-error')
    errorCount += checkInputElementIfEmpty('price-input-js', 'price-
error')
    errorCount += checkInputDate();
    errorCount += checkCategoriesCheckBoxes();

    if(errorCount){
        return;
    }

    let form = document.getElementById('insert-book-form');
    let formData = new FormData(form);

    $.ajax({
        url: 'models/book/add-book.php',
        type: 'POST',
        data: formData,
        contentType: false,
        processData: false,
        success: function(response) {
            clearAllInputBookValues();
            clearAllInputBookErrors();

```

```

        displayServerMessages('server-messages',response,'success-
message');
    },
    error: function(xhr, status, errorThrown) {
        let messages = JSON.parse(xhr.responseText);
        displayServerMessages('server-messages',messages,'error-
message');

        console.log(messages)
        console.log(xhr);
        console.log(status, errorThrown);
    }
});

}

function setSignupForm(){
    $(document).on('change', '#user-avatar-js',uploadAvatar);
    $(document).on('change', '#role-input-js',()=>{
        checkRole();
        setBiography();
    });
    $(document).on('blur', '#biography-input-js',checkBiography);

    $(document).on('blur', '#first-name-input-js',()=>{
        checkInputElementWithRegex(/^[A-ZŠĐĆŽ][a-zšđććž]{2,}([A-
ZŠĐĆŽ][a-zšđććž]{2,})*$/,'first-name-input-js','first-name-error');
    });
    $(document).on('blur', '#last-name-input-js',()=>{
        checkInputElementWithRegex(/^[A-ZŠĐĆŽ][a-zšđććž]{2,}([A-
ZŠĐĆŽ][a-zšđććž]{2,})*$/,'last-name-input-js','last-name-error');
    });
    $(document).on('blur', '#username-input-js',()=>{
        checkInputElementWithRegex(/^[a-zA-Z0-9.đžćčćšđž\(\)\/\-\_]{5,}$/,'username-input-js','username-error');
    });
    $(document).on('blur', '#password-input-js',()=>{
        checkInputElementWithRegex(/^(?=.*[a-zšđććž])(?=.*[A-
ZČĆŽĐ])(?=.*\d)(?=.*[_\(\)\/\-])[A-ZŠĐĆŽa-zšđććž\d._\(\)\/\-
]{5,})$/,'password-input-js','password-error');
    });
    $(document).on('blur', '#email-input-js',()=>{
        checkInputElementWithRegex(/^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-
]+\.[a-zA-Z]{2,}$/,'email-input-js','email-error');
    });
    $(document).on('blur', '#phone-input-js',()=>{
        checkInputElementWithRegex(/^d{5,15}$/, 'phone-input-
js','phone-error');
    });
}

```

```

    });
    $(document).on('blur', '#address-line-input-js', ()=>{
        checkInputElementWithRegex(/^[a-zšđžćčA-ZŠĐŽĆČ0-9\s\.\.-\#\\\/\,\,]+$/,'address-line-input-js','address-line-error');
    });
    $(document).on('blur', '#number-input-js', ()=>{
        checkInputElementWithRegex(/^d+[a-zA-Z]$/,'number-input-js','number-error');
    });
    $(document).on('blur', '#city-input-js', ()=>{
        checkInputElementWithRegex(/^([A-ZŠĐĆČ][a-zA-žđčć]{2,}\s?)+$/,'city-input-js','city-error');
    });
    $(document).on('blur', '#state-input-js', ()=>{
        checkInputElementWithRegex(/^([A-ZŠĐĆČ][a-zA-žđčć]{2,}\s?)+$/,'state-input-js','state-error');
    });
    $(document).on('blur', '#country-input-js', ()=>{
        checkInputElementWithRegex(/^([A-ZŠĐĆČ][a-zA-žđčć]{2,}\s?)+$/,'country-input-js','country-error');
    });
    $(document).on('blur', '#zip-code-input-js', ()=>{
        checkInputElementWithRegex(/^d{5,15}$/,'zip-code-input-js','zip-code-error');
    });

    $(document).on('click', '#create-account-button', sendSignupData);
}

function sendSignupData(){
    let errorCount = 0;

    let addressLine = document.getElementById('address-line-input-js').value;
    let addressNumber = document.getElementById('number-input-js').value;
    let city = document.getElementById('city-input-js').value;
    let state = document.getElementById('state-input-js').value;
    let zipCode = document.getElementById('zip-code-input-js').value;
    let country = document.getElementById('country-input-js').value;
    let role = document.getElementById('role-input-js').value;

    errorCount += checkAvatar();
    errorCount += checkInputElementWithRegex(/^[A-ZŠĐĆČ][a-zA-žđčć]{2,}([A-ZŠĐĆČ][a-zA-žđčć]{2,})*$/,'first-name-input-js','first-name-error');
    errorCount += checkInputElementWithRegex(/^[A-ZŠĐĆČ][a-zA-žđčć]{2,}([A-ZŠĐĆČ][a-zA-žđčć]{2,})*$/,'last-name-input-js','last-name-error');
}

```

```

        errorCount += checkInputElementWithRegex(/^[a-zA-Z0-
9.šđžćčĆŠĐŽ\(\)\/\-\_]{5,}$/, 'username-input-js', 'username-error');
        errorCount += checkInputElementWithRegex(/^(?=.*[a-zA-ZĆĆŽŽ])(?=.*[A-
ZĆĆŽŽĐĐ])(?=.*\d)(?=.*[_()]\-)[A-ZŠĐĆŽČa-zA-Zđććžđććžđ](_\()/\-
]{}$/,'password-input-js','password-error');
        errorCount += checkInputElementWithRegex(/^([a-zA-Z0-9._%+-]+@[a-zA-
Z0-9.-]+\.[a-zA-Z]{2,})$/,'email-input-js','email-error');
        errorCount += checkInputElementWithRegex(/^\\d{5,15}$/, 'phone-input-
js','phone-error');
        errorCount += checkRole();
        if(role == 3){
            errorCount += checkBiography();
        }

        if(addrLine.length || addresNumber.length || city.length ||
state.length || zipCode.length || country.length){
            errorCount += checkInputElementWithRegex(/^([a-zA-Zđććžđććž]-[A-ZŠĐĆĆŽ]-[0-
9]\s\.\-\#\\\/\,\,]+$/,'address-line-input-js','address-line-error');
            errorCount += checkInputElementWithRegex(/^\\d+[a-zA-
Z]?\$/,'number-input-js','number-error');
            errorCount += checkInputElementWithRegex(/^([A-ZŠĐĆĆŽ][a-
zđććž]{2,}\s?)+$/,'city-input-js','city-error');
            errorCount += checkInputElementWithRegex(/^([A-ZŠĐĆĆŽ][a-
zđććž]{2,}\s?)+$/,'state-input-js','state-error');
            errorCount += checkInputElementWithRegex(/^([A-ZŠĐĆĆŽ][a-
zđććž]{2,}\s?)+$/,'country-input-js','country-error');
            errorCount += checkInputElementWithRegex(/^\\d{5,15}$/, 'zip-
code-input-js','zip-code-error');
        }

        if(errorCount){
            return;
        }

        let form = document.getElementById('sign-up-form');
        let formData = new FormData(form);

        $.ajax({
            url: 'models/signup/signup.php',
            type: 'POST',
            data: formData,
            contentType: false,
            processData: false,
            success: function(response) {
                clearAllInputValues();
                clearAllInputErrors();

```

```

        displayServerMessages('server-messages',response,'success-
message');
    },
    error: function(xhr, status, errorThrown) {
        let messages = JSON.parse(xhr.responseText);
        displayServerMessages('server-messages',messages,'error-
message');

        console.log(messages)
        console.log(xhr);
        console.log(status, errorThrown);
    }
});

}

function cleareAllServerMessages(container){
    let serverMessagesDiv = document.querySelector(`.${container}`);
    while(serverMessagesDiv.firstChild)serverMessagesDiv.removeChild(se
rverMessagesDiv.firstChild);
}

function clearAllInputErrors(){
    document.getElementById('avatar-error').style.display = "none";
    document.getElementById('first-name-error').style.display = "none";
    document.getElementById('last-name-error').style.display = "none";
    document.getElementById('username-error').style.display = "none";
    document.getElementById('password-error').style.display = "none";
    document.getElementById('email-error').style.display = "none";
    document.getElementById('phone-error').style.display = "none";
    document.getElementById('biography-error').style.display = "none";

    document.getElementById('address-line-error').style.display =
"none";
    document.getElementById('number-error').style.display = "none";
    document.getElementById('city-error').style.display = "none";
    document.getElementById('state-error').style.display = "none";
    document.getElementById('zip-code-error').style.display = "none";
    document.getElementById('country-error').style.display = "none";
}

function clearAllInputValues(){
    document.getElementById('user-avatar-js').value = "";

    let avatarImg = document.getElementById('user-avatar-img');
    avatarImg.setAttribute('src','assets/images/avatars/default-
avatar.jpg');
}

```

```

document.getElementById('first-name-input-js').value = "";
document.getElementById('last-name-input-js').value = "";
document.getElementById('username-input-js').value = "";
document.getElementById('password-input-js').value = "";
document.getElementById('email-input-js').value = "";
document.getElementById('phone-input-js').value = "";
document.getElementById('biography-input-js').value = "";

document.getElementById('address-line-input-js').value = "";
document.getElementById('number-input-js').value = "";
document.getElementById('city-input-js').value = "";
document.getElementById('state-input-js').value = "";
document.getElementById('zip-code-input-js').value = "";
document.getElementById('country-input-js').value = "";
}

function displayServerMessages(container,messages,className){
    let serverMessagesDiv = document.querySelector(`#${container}`);
    serverMessagesDiv.style.display = 'block';

    while(serverMessagesDiv.firstChild)serverMessagesDiv.removeChild(se
rverMessagesDiv.firstChild);

    let ulTag = document.createElement('ul');

    messages.forEach(message => {
        let liTag = document.createElement('li');
        let pTag = document.createElement('p');
        pTag.classList.add(` ${className}`);
        pTag.textContent = message;

        liTag.appendChild(pTag);
        ulTag.appendChild(liTag);
    });

    serverMessagesDiv.appendChild(ulTag);
}

function checkInputElementWithRegex(regex,idElement,idErrorElement){
    let element = document.getElementById(`#${idElement}`).value;
    let errorMessage = document.getElementById(`#${idErrorElement}`);

    if(!regex.test(element)){
        errorMessage.style.display = 'block';
        return 1;
    }
}

```

```

        errorMessage.style.display = 'none';
        return 0;
    }

    function setBiography(){
        let role = document.getElementById('role-input-js').value;

        let title = document.getElementById('biography-title-js');
        let textarea = document.getElementById('biography-input-js');
        let errorMessage = document.getElementById('biography-error');

        if(role == 3){
            title.style.display = 'block';
            textarea.style.display = 'block';
            textarea.value = " ";
            errorMessage.style.display = 'none';
            return;
        }

        title.style.display = 'none';
        textarea.style.display = 'none';
        errorMessage.style.display = 'none';
    }

    function uploadAvatar(){

        let userAvatarImg = document.getElementById('user-avatar-img');
        let imageUploadInput = document.getElementById('user-avatar-js');

        if (imageUploadInput.files && imageUploadInput.files[0]
&& !checkAvatar()){

            let file = imageUploadInput.files[0];
            let fileURL = URL.createObjectURL(file);

            userAvatarImg.src = fileURL;
        }else{
            userAvatarImg.setAttribute('src','../assets/images/avatars/default-avatar.jpg');
        }
    }

    function checkBiography(){
        let biographyRegex = /(\s.*){4,}/;

        let biography = document.getElementById('biography-input-js').value;
        let errorMessage = document.getElementById('biography-error');
    }
}

```

```

        if(!biographyRegex.test(biography)){
            errorMessage.style.display = 'block';
            return 1;
        }

        errorMessage.style.display = 'none';
        return 0;
    }

function checkRole(){
    let role = document.getElementById('role-input-js').value;
    if(role == 3 || role == 2){
        return 0;
    }else{
        return 1;
    }
}

function checkAvatar(){

    let imageUploadInput = document.getElementById('user-avatar-js');
    let errorMessage = document.getElementById('avatar-error');

    if (imageUploadInput.files && imageUploadInput.files[0]){

        let file = imageUploadInput.files[0];

        let validExtensions = ["jpg", "jpeg", "png"];
        let fileExtension = file.name.split(".").pop().toLowerCase();

        if (!validExtensions.includes(fileExtension)){
            errorMessage.style.display = "block";
            return 1;
        }

        let fileSizeInKB = file.size / 1024;
        if (fileSizeInKB > 700) {
            errorMessage.style.display = "block";
            return 1;
        }
    }

    errorMessage.style.display = "none";
    return 0;
}

function setHeader(){
    $(document).on("click","#menu-icon-open",openPhoneNavigation);
}

```

```

$(document).on("click", "#menu-icon-close", closePhoneNavigation);
$(document).on("click", "#dark-mode-icon", toggleTheme);
$(document).on("click", "#light-mode-icon", toggleTheme);

    toggleThemeIcon();
}

function closePhoneNavigation(){

    let phoneNav = document.querySelector(".phone-nav");
    phoneNav.classList.remove("nav-entering");
    phoneNav.classList.add("nav-leaving");

    setTimeout(() => {
        $("body").removeClass("pre-loader");
        phoneNav.classList.remove("display-phone-nav");

    }, 450);
}

function toggleThemeIcon(){
    let light = document.getElementById("light-mode-icon");
    let dark = document.getElementById("dark-mode-icon");

    if(document.body.classList.contains("light-mode")){
        dark.style.display= "block";
        light.style.display = "none";
    }else{
        light.style.display= "block";
        dark.style.display = "none";
    }
}

function openPhoneNavigation(){
    let phoneNav = document.querySelector(".phone-nav");

    $("body").addClass("pre-loader");

    phoneNav.classList.add("display-phone-nav");
    phoneNav.classList.add("nav-entering");
    phoneNav.classList.remove("nav-leaving");
}

function toggleThemeIcon(){
    let light = document.getElementById("light-mode-icon");
    let dark = document.getElementById("dark-mode-icon");

```

```

        if(document.body.classList.contains("light-mode")){
            dark.style.display= "block";
            light.style.display = "none";
        }else{
            light.style.display= "block";
            dark.style.display = "none";
        }
    }

function toggleTheme(){
    let theme;

    if(document.body.classList.contains("light-mode")){
        document.body.classList.remove("light-mode");
        document.body.classList.add("dark-mode");
        toggleThemeIcon()
        theme = "dark";
    }else{
        document.body.classList.add("light-mode");
        document.body.classList.remove("dark-mode");
        toggleThemeIcon()
        theme = "light"
    }

   .setItemToLocalStorage("theme", theme);
}

function setThemeClass(){
    prefersDarkScheme = window.matchMedia("(prefers-color-scheme: dark)");
    let currentTheme = getItemFromLocalStorage("theme");

    if (currentTheme == "dark")
    {
        document.body.classList.toggle("dark-mode");
    }
    else if (currentTheme == "light")
    {
        document.body.classList.toggle("light-mode");
    }
    else if(prefersDarkScheme.matches){
        document.body.classList.toggle("dark-mode");
    }else{
        document.body.classList.toggle("light-mode");
    }
}

```

```
function getItemFromLocalStorage(name){
    let item =JSON.parse(localStorage.getItem(name));
    return item;
}

function setItemToLocalStorage(name,value){
    localStorage.setItem(name,JSON.stringify(value));
}

function ajaxCallback(file,method,data,result,error){

    if(method == 'GET'){
        $.ajax({
            url:file,
            method:'GET',
            dataType:'json',
            success:result,
            error:error
        })
    }

    if(method == 'POST'){
        $.ajax({
            url:file,
            method:'POST',
            dataType:'json',
            contentType: false,
            processData: false,
            data:data,
            success:result,
            error:error
        })
    }
}
```

_animations.scss

```
@keyframes changing-position-entering {
  0% {
    right: -120%;
  }
  100% {
    right: 0;
  }
}

@keyframes changing-position-leaving {
  0% {
    right: 0;
  }
  100% {
    right: -120%;
  }
}

@keyframes changing-blur {
  0% {
    opacity: 1;
    filter: blur(0px);
  }
  100% {
    opacity: 0.85;
    filter: blur(7px);
  }
}

@keyframes un-blur {
  0% {
    opacity: 0.85;
    filter: blur(7px);
  }
  100% {
    opacity: 1;
    filter: blur(0px);
  }
}
```

_bg-colors.scss

```
body{
    --bg-article-0:#cfbcbe;
    --bg-article-1:#f1e3e3;
    --bg-article-2:#cfdb6e0;
    --bg-article-3:#f1f0f0;
    --bg-article-4:#becab4;
    --bg-article-5:#dff0d8;
    --bg-article-6:#fcf1d9;
    --bg-article-7:#b7c9cc;
    --bg-article-8:#b9abb9;
    --bg-article-9:#e9e5c6;
}

body.dark-mode{
    --bg-article-0:hsl(0, 17%, 57%);
    --bg-article-1:hsl(0, 33%, 72%);
    --bg-article-2:hsl(215, 22%, 65%);
    --bg-article-3:hsl(0, 3%, 74%);
    --bg-article-4:hsl(93, 17%, 55%);
    --bg-article-5:hsl(103, 44%, 69%);
    --bg-article-6:hsl(41, 50%, 67%);
    --bg-article-7:hsl(189, 17%, 56%);
    --bg-article-8:hsl(300, 9%, 50%);
    --bg-article-9:hsl(53, 44%, 65%);
}

.bg-article-color-0{
    background: linear-gradient(to bottom, var(--bg-article-0) 85%, transparent 15%);
}

.bg-article-color-1{
    background: linear-gradient(to bottom, var(--bg-article-1) 85%, transparent 15%);
}

.bg-article-color-2{
    background: linear-gradient(to bottom, var(--bg-article-2) 85%, transparent 15%);
}

.bg-article-color-3{
```

```
background: linear-gradient(to bottom, var(--bg-article-3) 85%,  
transparent 15%);  
  
.bg-article-color-4{  
background: linear-gradient(to bottom, var(--bg-article-4) 85%,  
transparent 15%);  
  
.bg-article-color-5{  
background: linear-gradient(to bottom, var(--bg-article-5) 85%,  
transparent 15%);  
  
.bg-article-color-6{  
background: linear-gradient(to bottom, var(--bg-article-6) 85%,  
transparent 15%);  
  
.bg-article-color-7{  
background: linear-gradient(to bottom, var(--bg-article-7) 85%,  
transparent 15%);  
  
.bg-article-color-8{  
background: linear-gradient(to bottom, var(--bg-article-8) 85%,  
transparent 15%);  
  
.bg-article-color-9{  
background: linear-gradient(to bottom, var(--bg-article-9) 85%,  
transparent 15%);
```

_reset.scss

```
@import './variables';

*,::before,::after{
  box-sizing: border-box;
  scroll-behavior: smooth;
  padding: 0px;
  margin: 0px
}
:hover, :active{
  box-sizing: border-box;
}

*{
  font-family: $font1;
}
```

_responsive.scss

```
@mixin xxl{
  @media(max-width: 1400px){
    @content;
  }
}

@mixin xl{
  @media(max-width: 1200px){
    @content;
  }
}

@mixin lg{
  @media(max-width: 992px){
    @content;
  }
}

@mixin lg-940{
  @media(max-width: 940px){
    @content;
  }
}

@mixin md{
  @media(max-width: 768px){
    @content;
  }
}

@mixin xmd{
  @media(max-width: 650px){
    @content;
  }
}

@mixin sm{
  @media(max-width: 576px){
    @content;
  }
}

@mixin xs{
```

```
@media(max-width: 420px){  
    @content;  
}  
}
```

_variables.scss

```
$font1: 'Source Sans Pro';
$font2: 'Source Serif Pro';

$font-size:(
    'size-large':5rem,
    'size-big':3.3rem,
    'size-h1':2.5rem,
    'size-h2':2rem,
    'size-h3':1.75rem,
    'size-body':1.25rem,
    'size-small':1rem,
);

@function size($size-name){
    @return map-get($font-size , $size-name)
}
```

style.scss

```
@import './reset';
@import './responsive';
@import './variables';
@import './animations';
@import './bg-colors';

:root{
    //CONST
    --text-color-heading-const:#f5f5f5;
    --text-color-regular-const:#e0e0e0;

    --background-color-const: #00293f;
    --background-color-header-const: hsl(186, 59%, 30%);
    --background-color-hover-const:hsl(186, 62%, 22%);

    --text-shadow-const:rgb(0 0 0 / 70%);

    //DEFAULT LIGHT
    --text-color-heading:#0e0e0e;
    --text-color-regular:#1e1e1e;
    --text-color-green: hsl(186, 62%, 32%);

    --background-color: #fafafa;

    --star-orange: #e47b01;

    --gray-text-color: #596280;
    --gray-bg-color: rgba(255, 255, 255, 0);
    --gray-bg-color-hover:rgba(13, 13, 13, 0);

    --border-book-shadow: #e3e6ef;

    --bg--button:#fafafa;
    --bg-button-hover: hsl(184, 46%, 93%);
    --button-angle: #fafafa;

    --category-outline-hover:#b7bdc9;
    --category-outline:#cf6e0;

    --remove-button-bg: rgba(255, 255, 255, 0);
    --remove-button-bg-hover: rgb(224, 224, 224);

    --table-tr-bg:rgb(239, 239, 239);
    --table-tr-hover: #dfdfdfab;
```

```
}

.light-mode{
    --text-color-heading:#0e0e0e;
    --text-color-regular:#1e1e1e;
    --text-color-green: hsl(186, 62%, 32%);

    --background-color: #fafafa;

    --star-orange: #e47b01;

    --gray-text-color: #596280;
    --gray-bg-color: rgba(255, 255, 255, 0);
    --gray-bg-color-hover:rgba(13, 13, 13, 0);

    --border-book-shadow: #e3e6ef;

    --bg--button:#fafafa;
    --bg-button-hover: #e4f4f5;
    --button-angle: #fafafa;

    --category-outline-hover:#b7bdc9;
    --category-outline:#cf6e0;

    --remove-button-bg: rgba(255, 255, 255, 0);
    --remove-button-bg-hover: rgb(224, 224, 224);

    --table-tr-bg:rgb(239, 239, 239);
    --table-tr-hover: #dfdfdfab;
}

.dark-mode{
    --text-color-heading:#f5f5f5;
    --text-color-regular:#e0e0e0;
    --text-color-green: hsl(186, 62%, 42%);

    --background-color: #121212;

    --star-orange: #e38e2c;

    --gray-text-color: hsl(226, 18%, 63%);
    --gray-bg-color:#1b1b1b;
    --gray-bg-color-hover:hsl(0, 0%, 9%);

    --border-book-shadow: #e3e6ef00;

    --bg--button:#1b1b1b;
    --bg-button-hover: #052016;
    --button-angle: #fafafa;

    --category-outline-hover:hsla(186, 62%, 42%, 0);
}
```

```
--category-outline:hsla(186, 62%, 32%, 0);

--remove-button-bg: rgb(36, 36, 36);
--remove-button-bg-hover: rgb(12, 12, 12);

--table-tr-bg:rgb(28, 28, 28);
--table-tr-hover: #222222ab;

.svg{
    filter:brightness(80%);
}

.set-brightness{
    filter:brightness(85%);
}

.error-message{
    color: red;
}

.success-message{
    color: green;
}

.server-messages{
    padding-bottom: 2rem;

    li{
        margin-bottom:0.5rem;
    }
}

.text-bold{
    font-weight: 600;
}

.text-italic{
    font-style: italic;
}

a{
    text-decoration: none;
}

ul{
    list-style: none;
}

body{
    min-height: 100vh;
```

```
    max-height: fit-content;
    background-color: var(--background-color);
}
```

```
.pre-loader{
    overflow-y: hidden;
}
```

```
.wrapper{
    width: 1300px;
    margin: 0 auto;
```

```
@include xxl{
    width: 1100px;
}
```

```
@include xl{
    width: 100%;
    padding-left: 10px;
    padding-right: 10px;
}
}
```

```
.danger-option{
    color: rgb(255, 77, 77) !important;
    background-color: unset !important;
```

```
&:hover{
    color: hwb(0 13% 0%) !important;
    background-color: hsla(0, 100%, 70%, 0.267) !important;
}
```

```
.safe-option{
```

```
    background-color: var(--background-color-header-const) !important;
    color: var(--text-color-heading-const) !important;
```

```
&:hover{
    cursor: pointer;
    background-color: var(--background-color-hover-
const) !important;
}
```

```
.add-buttons-container{
    margin-bottom: 5rem;
```

```
// ----- HEADER -----
// -----



header{
    position: sticky;
    top: 0;
    height: 50px;
    background-color: var(--background-color-header-const);
    width: 100%;

    z-index: 10;

    display: grid;
    grid-template-columns: 1fr;
    grid-template-rows: 1fr;
}

#header{
    display: grid;
    grid-template-rows: 1fr;
    grid-template-columns: auto 1fr auto auto;

    grid-template-areas: "logo . links icons";



a{
    color: var(--text-color-heading-const);
    align-self: center;
    height: 50px;
    grid-area: logo;

    img{
        height: 50px;
    }
}

ul{
    display: grid;
    grid-auto-flow: column;
    grid-area: links;
    gap: 15px;

    @include xmd{
        display: none;
    }

    #active-link{
        &::after{
            width: 100%;
        }
    }
}
```

```
        }

    a{
        display: grid;
        align-items: center;

        font-size: size(size-body);

        position: relative;

        &::after{
            content: "";
            position: absolute;
            bottom: 0;
            left: 0;
            width: 0%;
            height: 4px;
            background-color: var(--text-color-heading-const);
            transition: width 0.3s;
        }

        &:hover{
            &::after{
                width: 100%;
            }
        }
    }

}

#icons-container{
    color: var(--text-color-heading-const);
    grid-area: icons;

    display: grid;
    align-items: center;
    grid-auto-flow: column;
    grid-auto-columns: 30px;
    gap: 15px;

    padding-left: 30px;

    font-size: size(size-body);

    // styling profile pic of an user
    div{

        img{
            height: 30px;
    
```

```
        border-radius: 50%;  
    }  
  
    width: 30px;  
    height: 30px;  
    display: block;  
    //background-color: red;  
    border-radius: 50%;  
  
    a{  
        display: block;  
        height: 100%;  
        width: 100%;  
    }  
}  
// end of styling profile pic of an user
```

```
i{  
    font-size: size(size-body);  
    &:hover{  
        cursor: pointer;  
    }  
}
```

```
#menu-icon-open{  
    display: none;  
  
    @include xmd{  
        display: block;  
    }  
}  
}
```

```
.phone-nav{  
    display: none;  
    position: fixed;  
    z-index: 16;  
    top: 0;  
    bottom: 0;  
    right: 0;  
    background-color: var(--background-color-header-const);  
}
```

```
i{  
    position: absolute;  
    top: 5px;  
    right: 10px;  
    color: white;
```

```
    font-size: size(size-h2);
```

```
    &:hover{
        cursor: pointer;
    }
}

.display-phone-nav{
    @include md{
        display: grid;
        width: 100%;

        align-items: center;
        justify-items: center;

        a{
            text-align: center;
            font-size: size(size-h3);
            color: var(--text-color-heading-const);
        }
    }

    li{
        display: grid;
        margin-bottom: 1rem;
    }
}

.nav-entering{
    right: 0;
    animation: changing-position-entering 0.5s;

}
.nav-leaving{
    right: -120%;
    animation: changing-position-leaving 0.5s;
}

// ----- HEADER END -----
// -----



// ----- FOOTER -----
// -----



footer{
    width: 100%;
    padding: 1rem 0;
```

```
    background-color: var(--background-color-header-const);  
}  
  
#footer{  
    //padding-top: 2rem;  
    display: grid;  
    grid-template-rows: 1fr;  
    grid-template-columns: max-content max-content max-content;  
    justify-content: space-between;  
    align-items: center;  
  
    @include md{  
        grid-template-columns: 1fr;  
        grid-template-rows: auto auto auto;  
        justify-content: center;  
        text-align: center;  
  
        gap: 1rem;  
    }  
  
    a{  
        font-size: size(size-body);  
        color: var(--text-color-heading-const);  
        font-weight: 600;  
  
        &:hover{  
            color: var(--text-color-regular-const);  
        }  
    }  
}
```

```
// ----- FOOTER END -----  
// -----
```

```
// ----- SIGN UP -----  
// -----
```

```
#sign-up-body{  
    min-height: 100vh;  
    max-height: fit-content;  
}  
  
.info-container-grid{  
    width: 100%;  
    height: auto;  
    margin-top: 2rem;
```

```
margin-bottom: 5rem;
display: grid;
grid-template-columns: 1fr 40px 1fr;
```

```
@include xmd{
    grid-template-columns: 1fr;
}
}
```

```
.input-title{
    font-size: size(size-body);
    margin-top: 1rem;
    margin-bottom: 0.3rem;
    font-weight: 500;
}
```

```
    color: var(--text-color-heading);
}
```

```
input,.user-information{
    height: 2rem;
    font-size: size(size-body);
}
```

```
.button-option-container{
    display: flex;
    justify-content: end;
```

```
input{
    height: unset;
    margin-left: 1rem;
```

```
    padding: 0.3rem 0.6rem;
    border: unset;
    border-radius: 3px;
    outline: none;
```

```
background-color: unset;
```

```
&:hover{
    cursor: pointer;
}
}
```

```
}
```

```
textarea{
    height: 100px;
    resize: none;
    font-size: size(size-body);
```

```
}

.sign-up-container{
    margin: 0 auto;
    margin-top: 5rem;
    margin-bottom: 5rem;
    width: 800px;
    height: auto;
    padding: 3rem 0.5rem;

    border-radius: 5px;
    border: 1px solid var(--border-book-shadow);
    background-color: var(--gray-bg-color);

@include lg-940{
    width: 100%;

}

h2{
    text-align: center;
    font-size: size(size-h2);
    margin-bottom: 3rem;
    color: var(--text-color-heading);

    span{
        font-size: size(size-body);
        font-weight: 400;
    }
}

.user-avatar-placeholder{
    margin: 0 auto;
    margin-bottom: 1rem;
    width: 150px;
    height: 150px;
    border-radius: 50%;
    overflow: hidden;

    img{
        width: 100%;
    }
}

.upload-avatar-text{
    text-align: center;
    margin-bottom: 0.5rem;
    font-size: size(size-body);
    height: auto;
    color: var(--text-color-regular);
```

```
        }

.user-avatar{
    margin: 0 auto;
    display: block;

    font-size: unset;
    font-weight: unset;
    color: var(--text-color-regular);

}

#avatar-error{
    margin: 0 auto;
    width: 300px;
    height: 100px;
    margin-top: 0.5rem;
    margin-bottom: 0.5rem;
}

.personal-info-container-grid{

    grid-template-rows: repeat(15,auto);
    // grid-template-rows: max-content max-content;

    grid-template-areas:
        "first-name-title . last-name-title"
        "first-name-input . last-name-input"
        "first-name-error . last-name-error"

        "username-title . password-title"
        "username-input . password-input"
        "username-error . password-error"

        "email-title email-title email-title"
        "email-input email-input email-input"
        "email-error email-error email-error"

        "phone-title . role-title"
        "phone-input . role-input"
        "phone-error . role-error"

        "biography-title biography-title biography-title"
        "biography-input biography-input biography-input"
        "biography-error biography-error biography-error"
        ;

@include xmd{
```

```
grid-template-rows: repeat(24,auto);
grid-template-areas:
"first-name-title"
"first-name-input"
"first-name-error"

"last-name-title"
"last-name-input"
"last-name-error"

"username-title"
"username-input"
"username-error"

"password-title"
"password-input"
"password-error"

"email-title"
"email-input"
"email-error"

"phone-title"
"phone-input"
"phone-error"

"role-title"
"role-input"
"role-error"

"biography-title"
"biography-input"
"biography-error"
;

}
```

```
.first-name-title{
    grid-area: first-name-title;
}

.first-name-input{
    grid-area: first-name-input;
}

#first-name-error{
    grid-area: first-name-error;
}
```

```
.last-name-title{  
    grid-area: last-name-title;  
}
```

```
.last-name-input{  
    grid-area: last-name-input;  
}
```

```
#last-name-error{  
    grid-area: last-name-error;  
}
```

```
.username-title{  
    grid-area: username-title;  
}
```

```
.username-input{  
    grid-area: username-input;  
}
```

```
#username-error{  
    grid-area: username-error;  
}
```

```
.password-title{  
    grid-area: password-title;  
}
```

```
.password-input{  
    grid-area: password-input;  
}
```

```
#password-error{  
    grid-area: password-error;  
}
```

```
.email-title{  
    grid-area: email-title;  
}
```

```
.email-input{  
    grid-area: email-input;  
}
```

```
#email-error{  
    grid-area: email-error;  
}
```

```
.phone-title{  
    grid-area: phone-title;  
}
```

```
.phone-input{  
    grid-area: phone-input;  
}
```

```
#phone-error{  
    grid-area: phone-error;  
}
```

```
.role-title{  
    grid-area: role-title;  
}
```

```
.role-input{  
    grid-area: role-input;  
}
```

```
#role-error{  
    grid-area: role-error;  
}
```

```
.biography-title{  
    grid-area: biography-title;  
}
```

```
.biography-input{  
    grid-area: biography-input;  
}
```

```
#biography-error{  
    grid-area: biography-error;  
}  
}
```

```
.address-info-container-grid{  
    grid-template-rows: repeat(12,auto);  
    grid-template-areas:  
        "address-line-title address-line-title address-line-title"  
        "address-line-input address-line-input address-line-input"  
        "address-line-error address-line-error address-line-error"  
        "number-title . .."  
        "number-input . .."  
        "number-error . .."
```

```
"city-title . state-title"
"city-input . state-input"
"city-error . state-error"

"zip-code-title . country-title"
"zip-code-input . country-input"
"zip-code-error . country-error";
```

```
@include xmd{
    grid-template-rows: repeat(18,auto);
    grid-template-areas:
        "address-line-title"
        "address-line-input"
        "address-line-error"

        "number-title"
        "number-input"
        "number-error"

        "city-title"
        "city-input"
        "city-error"
```

```
"state-title"
"state-input"
"state-error"

"zip-code-title"
"zip-code-input"
"zip-code-error"
```

```
"country-title"
"country-input"
"country-error";
```

```
}
```

```
.address-line-title{
    grid-area: address-line-title;
}
```

```
.address-line-input{
    grid-area: address-line-input;
}
```

```
#address-line-error{
    grid-area: address-line-error;
}
```

```
.number-title{  
    grid-area: number-title;  
}
```

```
.number-input{  
    grid-area: number-input;  
}
```

```
#number-error{  
    grid-area: number-error;  
}
```

```
.city-title{  
    grid-area: city-title;  
}
```

```
.city-input{  
    grid-area: city-input;  
}
```

```
#city-error{  
    grid-area: city-error;  
}
```

```
.state-title{  
    grid-area: state-title;  
}
```

```
.state-input{  
    grid-area: state-input;  
}
```

```
#state-error{  
    grid-area: state-error;  
}
```

```
.zip-code-title{  
    grid-area: zip-code-title;  
}
```

```
.zip-code-input{  
    grid-area: zip-code-input;  
}
```

```
#zip-code-error{  
    grid-area: zip-code-error;  
}
```

```
.country-title{  
    grid-area: country-title;  
}
```

```
.country-input{
    grid-area: country-input;
}

#country-error{
    grid-area: country-error;
}
```

```
select{
    height: 2rem;
    font-size: size(size-body);
}
```

```
#create-account-button{
    display: block;
    height: auto;
    margin: 0 auto;
    padding: 0.3rem 0.6rem;
    border: unset;
    border-radius: 3px;
    outline: none;

background-color: var(--background-color-header-const);
color: var(--text-color-heading-const);
```

```
&:hover{
    cursor: pointer;
    background-color: var(--background-color-hover-const);
}
```

```
p{
    height: auto !important;
}
```

```
// ----- SIGN UP END -----
// -----
```

```
// ----- LOG IN -----
// -----
```

```
#log-in-section{
    width: 100%;
```

```
    min-height: calc(100vh - 50px);
    max-height: fit-content;
}
```

```
#log-in-section{
    display: grid;
    align-items: center;
    justify-content: center;
}
```

```
#login-form{
    width: 300px;
    padding: 3rem 0.5rem;

    border-radius: 5px;
    border: 1px solid var(--border-book-shadow);
    background-color: var(--gray-bg-color);
```

```
h2{
    text-align: center;
    font-size: size(size-h2);
    margin-bottom: 3rem;
    color: var(--text-color-heading);
}
```

```
#log-in-container{
    width: 100%;
    padding-bottom: 1rem;
    display: grid;
    grid-template-columns: 1fr;
    grid-template-rows: auto auto auto auto auto auto;
```

```
grid-template-areas:
    "username-title"
    "username-input"
    "username-error"
    "password-title"
    "password-input"
    "password-error";
```

```
#username-title{
    grid-area: username-title;
}
```

```
#sername-input-js{
    grid-area: sername-input;
}
```

```
#username-error{
```

```
        grid-area: username-error;
    }

#password-title{
    grid-area: password-title;
}

#password-input-js{
    grid-area: password-input;
}

#password-error{
    grid-area: password-error;
}

.input-title{
    font-size: size(size-body);
    margin-top: 1rem;
    margin-bottom: 0.3rem;
    font-weight: 500;
}

    color: var(--text-color-heading);
}

input{
    height: 2rem;
    font-size: size(size-body);
}

#log-in-button{
    display: block;
    height: auto;
    margin: 0 auto;
    padding: 0.3rem 0.6rem;
    border: unset;
    border-radius: 3px;
    outline: none;
}

background-color: var(--background-color-header-const);
color: var(--text-color-heading-const);

&:hover{
    cursor: pointer;
    background-color: var(--background-color-hover-const);
}

}

// ----- LOG IN END -----
```

```
// -----  
// ----- USER PROFILE -----  
// -----  
  
#user-profile-section{  
    width: 100%;  
    min-height: calc(100vh - 50px);  
    max-height: auto;  
  
    .user-information{  
        font-weight: 600;  
        font-size: 1.4rem;  
        color: var(--text-color-heading);  
    }  
  
    .input-title{  
        color: var(--text-color-regular);  
    }  
}  
  
-----
```

```
#writter-option-wrapper{  
    width: 800px;  
    margin: 0 auto;  
    margin-bottom: 5rem;  
  
    @include lg-940{  
        width: 100%;  
        padding-left: 0.5rem;  
    }  
  
    .admin-option-container{  
        grid-template-columns: 1fr 1fr;  
        gap: 20px;  
  
        @include xl{  
            grid-template-columns: 1fr 1fr;  
        }  
  
        @include xmd{  
            grid-template-columns: 1fr;  
        }  
    }  
}
```

```
        }
    }
```

```
// ----- USER PROFILE END -----
// -----
```

```
// ----- ADMIN -----
// -----
```

```
#admin-section{
    min-height: calc(100vh - 50px);
    max-height: auto;
    padding: 5rem 0;
```

```
    @include xl{
        padding: 5rem 1rem;
    }
```

```
}
```

```
.admin-option-title{
    margin-bottom: 1rem;
    font-size: size(size-h2);
    color: var(--text-color-heading);
}
```

```
.admin-option-container{
    width: 100%;
    height: auto;
    margin-bottom: 3rem;
```

```
    display: grid;
    grid-auto-rows: 120px;
    grid-template-columns: 1fr 1fr 1fr;
    gap: 20px;
```

```
    @include xl{
        grid-template-columns: 1fr 1fr;
    }
```

```
    @include xmd{
        grid-template-columns: 1fr;
    }
}
```

```
.option-name-container{
```

```
    display: grid;
    align-content: center;
    padding-left: 1rem;
```

```
p{
    font-weight: 600;
    font-size: size(size-body);
    color: var(--text-color-regular);
}
}
```

```
.option-icon-container{
    background-color: var(--bg-article-7);
    display: grid;
    justify-content: center;
    align-items: center;
    border-radius: 5px;
    img{
        height: 85px;
    }
}
```

```
.admin-option{
    background-color: var(--gray-bg-color);
    border: 1px solid var(--border-book-shadow);
    position: relative;
    border-radius: 5px;

    display: grid;
    grid-template-rows: 1fr;
    grid-template-columns: 1fr 120px;
```

```
a{
    position: absolute;
    top: 0;
    bottom: 0;
    right: 0;
    left: 0;
}
```

```
&:hover{
    background-color: var(--gray-bg-color-hover);
    box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
}
```

```
// ----- ADMIN END -----
```

```
// -----  
  
// ----- ADMIN USERS -----  
// -----  
.table-container{  
    overflow-x: scroll;  
    margin-bottom: 5rem;  
  
    table{  
        border-collapse: collapse;  
        border-spacing: 0;  
        width: 100%;  
    }  
  
    thead{  
        height: 2rem;  
        background-color: var(--background-color-header-const);  
  
        font-size: size(size-body);  
  
        width: 100%;  
  
        th{  
            padding-left: 0.5rem;  
            font-weight: 600;  
            text-align: left;  
            color: var(--text-color-heading-const);  
        }  
  
    }  
  
    tbody{  
        tr{  
            &:hover{  
                background-color: var(--table-tr-hover) !important;  
            }  
        }  
  
        td{  
            height: 3rem;  
            padding-left: 0.5rem;  
            font-size: size(size-body);  
            color: var(--text-color-regular);  
  
            input[type=button]{  
                padding: 0.2rem 0.4rem;  
                border-radius: 5px;  
  
                font-size: size(size-body);  
            }  
        }  
    }  
}
```

```
        border: unset;
        border-radius: 3px;
        outline: none;

        background-color: unset;

        &:hover{
            cursor: pointer;
        }
    }
```

```
a{
    padding: 0.2rem 0.4rem;
    border-radius: 5px;
}
}
}
}
```

```
#admin-users-section{
    min-height: calc(100vh - 50px);
    padding: 5rem 0;
    @include xl{
        padding: 5rem 1rem;
    }
}
```

```
h2{
    margin-bottom: 1rem;
    color: var(--text-color-heading);
    font-size: size(size-h2);
}
```

```
table{
    @include xl{
        width: 1200px;
    }
}
```

```
#admin-users-server-messages{
    display: none;
    padding-top: 5rem;
    padding-bottom: 0;
}
// ----- ADMIN USERS END -----
// -----
```

```
// ----- ADMIN PUBLISHERS -----
```

```
// -----  
  
.insert-update-row-form{  
    width: 300px;  
    margin: 0 auto;  
    padding: 2rem 0.5rem;  
    border-radius: 5px;  
    border: 1px solid var(--border-book-shadow);  
    background-color: var(--gray-bg-color);  
    color: var(--text-color-heading);  
  
label{  
    display: block;  
    width: 100%;  
    font-size: size(size-body);  
    margin-bottom: 0.3rem;  
    font-weight: 500;  
}  
  
input, select{  
    width: 100%;  
    height: 2rem;  
    font-size: size(size-body);  
  
}  
  
select{  
    margin-bottom: 1rem;  
}  
  
.button-option-container{  
    margin-top: 1.5rem;  
    display: flex;  
    justify-content: end;  
  
input{  
    width: unset;  
    margin-left: 1rem;  
  
    padding: 0.3rem 0.6rem;  
    border: unset;  
    border-radius: 3px;  
    outline: none;  
    color: var(--text-color-regular);  
    background-color: unset;  
  
    &:hover{  
        cursor: pointer;  
    }  
}
```

```
        }
    }

#admin-publishers-section{
    min-height: calc(100vh - 50px);
    padding-top: 5rem;
    padding-bottom: 5rem;

h2{
    margin-bottom: 1rem;
    color: var(--text-color-heading);
    font-size: size(size-h2);
}

table{
    @include xl{
        width: 1200px;
    }
}

.add-button-container{
    display: flex;
    justify-content: end;
    padding-right: 0.5rem;

input,a{
    padding: 0.2rem 0.4rem;
    border-radius: 5px;
    margin-left: 1rem;
    font-size: size(size-body);

    border: unset;
    border-radius: 3px;
    outline: none;

    background-color: unset;

    &:hover{
        cursor: pointer;
    }
}

// ----- ADMIN PUBLISHERS END -----
// -----
```

```
// ----- ADMIN SURVEY -----  
// -----
```

```
#admin-survey-section{  
    min-height: calc(100vh - 50px);  
    padding-top: 5rem;  
    padding-bottom: 5rem;  
  
h2{  
    margin-bottom: 1rem;  
    color: var(--text-color-heading);  
    font-size: size(size-h2);  
}  
  
table{  
    @include xl{  
        width: 1200px;  
    }  
}  
  
.survey-table-options{  
    text-align: center;  
}  
}
```

```
// ----- ADMIN SURVEY END -----  
// -----
```

```
// ----- ADMIN CATEGORIES -----  
// -----
```

```
#admin-categories-section{  
    min-height: calc(100vh - 50px);  
    padding: 5rem 0;  
  
h2{  
    margin-bottom: 1rem;  
    color: var(--text-color-heading);  
    font-size: size(size-h2);  
}  
  
table{  
    @include xl{  
        width: 1200px;  
    }  
}  
}
```

```
// ----- ADMIN CATEGORIES END -----  
// -----
```

```
// ----- ADMIN MESSAGES -----  
// -----
```

```
#admin-messages-section{  
    min-height: calc(100vh - 50px);  
    padding-top: 5rem;  
    padding-bottom: 5rem;  
  
h2{  
    margin-bottom: 1rem;  
    color: var(--text-color-heading);  
    font-size: size(size-h2);  
}
```

```
table{  
    @include xl{  
        width: 1200px;  
    }  
}
```

```
.td-text{  
    width: 500px;  
    padding: 0.5rem;  
}  
}
```

```
// ----- ADMIN MESSAGES END -----  
// -----
```

```
// ----- WRITER BOOKS -----  
// -----
```

```
#writer-books-section{  
    min-height: calc(100vh - 50px);  
    padding: 5rem 0;  
  
h2{  
    margin-bottom: 1rem;  
    color: var(--text-color-heading);  
    font-size: size(size-h2);  
}
```

```
table{  
    @include xl{  
        width: 1200px;  
    }  
}
```

```
}

// ----- WRITER BOOKS END -----
// -----



// ----- INSERT UPDATE BOOK -----
// -----



.insert-update-book-section{
    min-height: calc(100vh - 50px);
    max-height: fit-content;
}

.td-image{
    padding: 1rem;
    img{
        height: 100px;
    }
}

.book-form-container{
    margin: 0 auto;
    margin-top: 5rem;
    margin-bottom: 5rem;
    width: 800px;
    height: auto;
    padding: 3rem 0.5rem;

    border-radius: 5px;
    border: 1px solid var(--border-book-shadow);
    background-color: var(--gray-bg-color);

    @include lg-940{
        width: 100%;
    }
}

h2{
    text-align: center;
    font-size: size(size-h2);
    margin-bottom: 3rem;
    color: var(--text-color-heading);

}

.book-image-container-form{
    width: 180px;
    height: 220px;
    margin: 0 auto;
    margin-bottom: 2rem;
```

```
        background-color: var(--bg-article-7);  
    }  
  
.upload-book-image-text{  
    text-align: center;  
    margin-bottom: 0.5rem;  
    font-size: size(size-body);  
    height: auto;  
    color: var(--text-color-regular);  
}  
  
.book-image-form{  
    margin: 0 auto;  
    display: block;  
  
    font-size: unset;  
    font-weight: unset;  
    color: var(--text-color-regular);  
}  
  
#book-image-error{  
    margin: 0 auto;  
    width: 300px;  
    height: 100px;  
    margin-top: 0.5rem;  
    margin-bottom: 0.5rem;  
}  
  
.book-info-container-grid{  
  
    grid-template-rows: repeat(15,auto);  
    // grid-template-rows: max-content max-content;  
  
    grid-template-areas:  
    "book-title-title book-title-title book-title-title"  
    "book-title-input book-title-input book-title-input"  
    "book-title-error book-title-error book-title-error"  
  
    "page-count-title . price-title"  
    "page-count-input . price-input"  
    "page-count-error . price-error"  
  
    "release-date-title . publisher-title"  
    "release-date-input . publisher-input"  
    "release-date-error . publisher-error"  
  
    "book-description-title book-description-title book-  
description-title"
```

```
"book-description-input book-description-input book-
description-input"
    "book-description-error book-description-error book-
description-error"

"book-category-title . ."
"book-category-input . ."
"book-category-error . ."
;

@include xmd{

grid-template-rows: repeat(21,auto);
grid-template-areas:
"book-title-title"
"book-title-input"
"book-title-error"

"page-count-title"
"page-count-input"
"page-count-error"

"price-title"
"price-input"
"price-error"

"release-date-title"
"release-date-input"
"release-date-error"

"publisher-title"
"publisher-input"
"publisher-error"

"book-description-title"
"book-description-input"
"book-description-error"

"book-category-title "
"book-category-input "
"book-category-error "
;
}

.book-description-title{
    grid-area: book-description-title;
}

.book-description-input{
    grid-area: book-description-input;
}
```

```
}

#book-description-error{
    grid-area: book-description-error;
}

.book-title-title{
    grid-area: book-title-title;
}

.book-title-input{
    grid-area: book-title-input;
}

#book-title-error{
    grid-area: book-title-error;
}

.page-count-title{
    grid-area: page-count-title;
}

.page-count-input{
    grid-area: page-count-input;
}

#page-count-error{
    grid-area: page-count-error;
}

.price-title{
    grid-area: price-title;
}

.price-input{
    grid-area: price-input;
}

#price-error{
    grid-area: price-error;
}

.release-date-title{
    grid-area: release-date-title;
}

.release-date-input{
    grid-area: release-date-input;
```

```
}

#release-date-error{
    grid-area: release-date-error;
}

.publisher-title{
    grid-area: publisher-title;
}

.publisher-input{
    grid-area: publisher-input;
}

#publisher-error{
    grid-area: publisher-error;
}

.book-category-title{
    grid-area: book-category-title;
}

.book-category-input{
    grid-area: book-category-input;
}

#book-category-error{
    grid-area: book-category-error;
}

.book-category-input{

    color: var(--text-color-regular);
    p{
        i{
            margin-right: 0.5rem;
        }
    }

    border: 1px solid var(--border-book-shadow);
    background-color: var(--gray-bg-color);
    padding: 0.3rem;
    border-radius: 5px;

    &:hover{
        cursor: pointer;
    }
}
```

```
        }
    }
    font-size: size(size-body);

.book-category-li{

    padding: 0 1.5rem;
    display: grid;
    grid-template-columns: 20px 1fr;
    gap: 5px;
    grid-template-rows: 1fr;
    align-items: center;

    label{
        &:hover{
            cursor: pointer;
        }
    }
}

select{
    height: 2rem;
    font-size: size(size-body);

@include lg-940{
    max-width: 300px;

}
option{
    width: 220px;
    overflow-wrap: break-word;
    word-wrap: break-word;
}

}

.book-image-container-form{
    display: grid;
    align-items: center;
    justify-content: center;

    img{
        height: 200px;
    }
}
```

```
// ----- INSERT UPDATE BOOK END -----
// ----- HOME -----



#about-readily{
    display: grid;
    justify-content: center;
    height: auto;
    padding-top: 3rem;

    background-color: var(--background-color-const);

    text-align: center;
    text-shadow: 0 2px 4px var(--text-shadow-const);

h1{
    color: var(--text-color-heading-const);
    font-size: size(size-big);
    font-weight:600;
    margin-bottom: 2rem;

}

#about-container{
    width: 600px;
    justify-self: center;
    margin-bottom: 5rem;

p{
    color: var(--text-color-regular-const);
    font-size: size(size-body);
}

@include xmd{
    width: 100%;
}

}

#img-placeholder{
    justify-self: center;
    width: 850px;
    //background-color: rgb(63, 26, 26);

    // podesiti height na auto kad budem stavio sliku
    height: auto;
    position: relative;
    img{
        width: 100%;
```

```
        }

    @include lg-940{
        width: 600px;
    }

    @include xmd{
        width: 550px;
    }

    @include sm{
        width: 300px;
    }
}

#angle-down{
    position: relative;
    margin-bottom: 3rem;
    padding-top: 1rem;
    display: grid;
    justify-content: center;
    align-items: center;

    font-size: size(size-h3);
    font-weight: 800;

    #position{
        position: absolute;
        top: -45px;
    }

    a{
        color: var(--text-color-regular);

    }
}

#scroll-down{
    &:hover{
        cursor: pointer;
    }
}

#introduction{
    margin-bottom: 3rem;
    display: grid;

    grid-template-rows: auto auto;
    grid-template-columns: 1fr 1fr 1fr;
    gap: 40px;
}
```

```
grid-template-areas: "heading heading heading"  
                      "article1 article2 article3";
```

```
@include md{  
    grid-template-columns: 1fr;  
    grid-template-rows: auto auto auto auto;  
    grid-template-areas: "heading"  
                        "article1"  
                        "article2"  
                        "article3";  
}
```

```
h3,h2{  
    color: var(--text-color-heading);  
}  
p{  
    margin-bottom: 3rem;  
    color: var(--text-color-regular);  
}
```

```
#introduction-heading-container{  
    grid-area: heading;
```

```
    text-align: center;  
    font-size: size(size-h3);
```

```
    br{  
        @include md{  
            display: none;  
        }  
    }  
}
```

```
article{
```

```
    h3{  
        margin-bottom: 1rem;  
    }  
}
```

```
#article1{  
    grid-area: article1;  
}
```

```
#article2{  
    grid-area: article2;  
}
```

```
#article3{  
    grid-area: article3;
```

```
        }
    }

#discover{
    position: relative;
    height: 300px;
    width: 100%;

    background-color: var(--background-color-const);

    display: grid;
    justify-content: center;

#discover-container{
    text-align: center;
    h2,a{
        color: var(--text-color-heading-const);
    }

p{
    color: var(--text-color-regular-const);
}

h2{
    margin-top: 2rem;
    margin-bottom: 1.5rem;
    font-size: size(size-h2);
}
p{
    font-size: size(size-body);
    margin-bottom: 3rem;
}

a{
    //border: 2px solid white;
    font-weight: 600;
    background-color: var(--background-color-header-const);
    // color: var(--text-color-heading);
    padding: 10px;
    border-radius: 3px;
    font-size: size(size-h3);

    &:hover{
        background-color: var(--background-color-hover-const);
    }
}

#discover-img-1{
    position: absolute;
```

```
        bottom: -30px;  
        left: 0;  
        height: 200px;  
    }  
  
#discover-img-2{
```

```
    position: absolute;  
    bottom: -30px;  
    right: 0;  
    height: 200px;  
}  
}
```

```
#suggested-books{  
    margin-top: 5rem;  
}
```

```
.section-articles{  
    position: relative;  
    margin-bottom: 5rem;
```

```
@include xl{  
    width: 85%;  
    margin: 0 auto;  
    margin-bottom: 5rem;  
}
```

```
h2{  
    font-family: $font2;  
    color: var(--text-color-heading);  
    margin-bottom: 2rem;  
}  
}
```

```
.article-container{  
    overflow-x: auto;  
  
    display: grid;  
    grid-auto-flow: column;  
  
    grid-template-rows: auto;  
  
    &::-webkit-scrollbar{  
        display: none;  
    }  
}  
  
.article-book-container{
```

```
    grid-auto-columns: 180px;
    gap: 40px;
}

.dark-mode .article-category-container{
    article{
        &:hover{
            border: 1px solid transparent;
            padding: 10px !important;
        }
    }
}

.article-category-container{
    grid-auto-columns: 250px;
    gap: 20px;
    //outline: 1px solid black;

}

.article-category{
    border: 1px solid var(--category-outline);
    //outline-offset: -2px;
    background-color: var(--gray-bg-color);

    border-radius: 3px;

    height: 85px;
    position: relative;
    overflow: hidden;

    display: grid;
    grid-template-rows: 1fr;
    grid-template-columns: auto max-content;
    gap: 20px;

    padding: 10px;
    font-weight: 600;
    color: var(--text-color-heading);

    &:hover{
        //outline-color: var(--gray-bg-color-hover);
        padding: 9px;
        border: 2px solid var(--category-outline-hover);

        background-color: var(--gray-bg-color-hover);
    }
}
```

```
a{  
    z-index: 2;  
    position: absolute;  
    top: 0;  
    bottom: 0;  
    right: 0;  
    left: 0;  
}
```

```
img{  
    z-index: 1;  
    width : 80px;  
    box-shadow: 0 6px 15px rgb(0 0 0 / 15%);  
    transform: rotate(18deg);  
    border-radius: 2px;  
    position: relative;  
    top: 10px;  
    right: 20px;  
}  
}
```

```
.article-author-container{  
    grid-auto-columns: 180px;  
    gap: 40px;  
}
```

```
.article-div-img-container{  
    //outline: 2PX solid rebeccapurple;  
    border-radius: 3px 3px 0 0 ;  
    width: 100%;  
    height: 220px;  
    display: grid;  
    justify-content: center;  
    align-items: end;
```

```
img{  
    height: 200px;  
    max-width: 170px;  
    box-shadow: 0 6px 6px rgb(0 0 0 / 35%);  
}  
}
```

```
.article-book{  
    display: grid;  
    grid-template-columns: 1fr;  
    grid-template-rows: max-content auto;
```

```
    position: relative;
```

```
border: 1px solid var(--border-book-shadow);  
background-color: var(--gray-bg-color);
```

```
&:hover > .link-to-single-a-book{  
    box-shadow: 0 2px 10px rgb(0 0 0 / 10%);  
}
```

```
&:hover{  
    background-color: var(--gray-bg-color-hover);  
}
```

```
h3{  
    font-weight: 600;  
    color: var(--text-color-heading);  
    font-size: size(size-body);  
    line-height: 1.3;
```

```
//margin-top: 0.5rem;  
// margin-bottom: ;  
}
```

```
.author-link{  
    font-size: size(size-small);  
    line-height: 1.5;  
    color: var(--text-color-green);  
    font-weight: bold;
```

```
position: relative;  
z-index: 4;  
padding: 3px 0;
```

```
&:hover{  
    color: var(--background-color-hover-const);  
}  
}
```

```
.link-to-single-a-book{  
    position: absolute;  
    top: 0;  
    bottom: 0;  
    left: 0;  
    right: 0;
```

```
    z-index: 3;  
}
```

```
.title-and-author{  
    padding-top: 1rem;  
    padding-bottom: 0.5rem;
```

```
}
```

```
.article-books-text-container{  
    display: grid;  
    grid-template-columns: 1fr;  
    grid-template-rows: max-content max-content;  
    align-content: space-between;  
  
    padding: 0.2rem;  
}
```

```
.stars-and-cart-container{  
    display: grid;  
    grid-template-columns: auto auto;  
    grid-template-rows: auto auto;  
    grid-template-areas:    "stars cart"  
                        "review cart";
```

```
.shopping-cart{  
    justify-self: center;  
    align-self: center;  
    grid-area: cart;
```

```
    position: relative;  
    z-index: 4;
```

```
    color: var(--text-color-regular);
```

```
&:hover{  
    cursor: pointer;
```

```
        color: var(--background-color-header-const);
```

```
    }  
}
```

```
.stars-container{  
    grid-area: stars;
```

```
i{  
    color: var(--star-orange);  
}  
}
```

```
.rating-text{  
    grid-area: review;  
    color: var(--gray-text-color);  
    font-size: size(size-small);  
    font-weight: 600;
```

```
        }
    }

.angle-container{
    position: absolute;
    z-index: 5;

    width: 40px;
    height: 40px;
    border-radius: 50%;
    border: 1px solid var(--text-color-green);
    background-color: var(--bg--button);

    display: grid;
    justify-content: center;
    align-items: center;

    box-shadow: 0 3px 6px rgb(0 0 0 / 20%);

    &:hover{
        cursor: pointer;
        border: 2px solid var(--background-color-header-const) ;
        background-color: var(--bg-button-hover);
    }

    i{
        font-size: 20px;
        color: var(--text-color-green);
    }
}

// ----- HOME END -----
// -----
```



```
//----- WRITER -----
//-----
```



```
#writer-info{
    width: 100%;

    background-color: var(--background-color-const);
}

#writer-info-container{
    display: grid;
    padding: 2rem;
    grid-template-columns: auto 1fr;
    grid-template-rows: auto auto;
    grid-template-areas: "writer-img heading"
```

```
"writer-img writer-info";
```

```
gap:20px;
```

```
@include md{  
    grid-template-columns: auto;  
    grid-template-rows: auto auto auto;  
    grid-template-areas: "writer-img"  
                        "heading"  
                        "writer-info";
```

```
justify-content: center;  
align-items: center;
```

```
img{  
    justify-self: center;  
}
```

```
h1{  
    text-align: center;  
}  
}
```

```
img{  
    filter:brightness(85%);  
    height: 100px;  
    grid-area: writer-img;  
    border-radius: 50%;  
    border: 3px solid var(--text-color-heading-const);  
}
```

```
h1{  
    grid-area: heading;  
    color: var(--text-color-heading-const);  
}
```

```
p{  
    grid-area: writer-info;  
    color: var(--text-color-regular-const);  
    font-size: size(size-body);  
}
```

```
span{  
    font-weight: 600;  
    margin-left: 10px;  
    color: var(--text-color-heading-const);  
    text-decoration: underline;  
  
    &:hover{  
        cursor: pointer;
```

```
        }
    }
}

#writer-books{
    padding-top: 5rem;
}

.article-writer{
    position: relative;
    height: auto;
    padding-top: 0.3rem;
    padding-bottom: 0.3rem;
    display: grid;
    grid-auto-columns: auto;
    grid-template-rows: auto auto auto;
    justify-content: center;
    align-items: center;
    border: 1px solid var(--border-book-shadow);
    text-align: center;
    background-color: var(--gray-bg-color);
}

h3{
    color: var(--text-color-heading);
}

p{
    color: var(--gray-text-color);
}

&:hover{
    box-shadow: 0 2px 10px rgb(0 0 0 / 10%);
    background-color: var(--gray-bg-color-hover);
}

a{
    position: absolute;
    z-index: 4;
    top: 0;
    bottom: 0;
    left: 0;
    right: 0;
}
}

.article-writer-img{
    justify-self: center;
    height: 75px;
```

```
border-radius: 50%;  
}  
  
//----- WRITER END -----  
//-----  
  
//----- BOOK -----  
//-----  
  
//----- BOOK.PHP -----  
//-----  
  
#book-info{  
    padding-top: 4rem;  
    display: grid;  
    grid-template-rows: auto auto auto auto auto auto;  
    grid-template-columns: 700px 350px;  
    justify-content: space-between;  
    // height: 700px;  
    //overflow: hidden;  
  
    grid-template-areas:  
        "heading img-container"  
        "author img-container"  
        "stars img-container"  
        "about-book img-container"  
        "related-categories img-container"  
        "info img-container";  
  
    @include xl{  
        grid-template-rows: auto auto auto auto auto auto;  
        grid-template-columns: 100%;  
  
        justify-content: space-between;  
        // height: 700px;  
  
        grid-template-areas:  
            "img-container"  
            "heading"  
            "author"  
            "stars"  
            "about-book"  
            "related-categories"  
            "info";  
    }  
}
```

```
#div-info-section{
  h2{
    display: none;
  }

  //margin-top: 3rem;
  margin-bottom: 0;
  //padding-bottom: 0;

  grid-area: info;
}

#div-info-container{
  p{
    margin-bottom: 0.3rem;
    font-size: size(size-body);
  }
}

.heading-info{
  font-size: size(size-small);
  color: var(--gray-text-color);
}

#container-div-image-and-price{
  position: sticky;
  top: 50px;
  grid-area: img-container;
  height: fit-content;
}

@include xl{
  position: relative;
  top: unset;

  width: 500px;
  justify-self: center;
  margin-bottom: 3rem;
}

@include sm{
  width: 100%;
}

border: 1px solid var(--category-outline);
border-radius: 3px;
background-color: var(--gray-bg-color);

#container-div-img{
  display: grid;
}
```

```
    justify-content: center;
    align-items: end;
    height: 440px;
```

```
    }
}
```

```
#price-container{
    padding: 1rem;
```

```
    display: grid;
    grid-auto-flow: column;
    justify-content: space-between;
    align-items: center;
```

```
    font-size: size(size-h3);
    color: var(--text-color-heading);
    p{
        font-weight: 600;
    }
```

```
    i{
        &:hover{
            cursor: pointer;
            color: var(--text-color-green);
        }
    }
}
```

```
img{
    height: 400px;
```

```
// @include sm{
//     width: 300px;
// }
}
```

```
#text-container-about{
    grid-area: about-book;
    margin-bottom: 3rem;
```

```
h2{
    color: var(--text-color-heading);
    font-family: $font2;
    margin-bottom: 1.5rem;
    font-size: size(size-h3);
}
```

```
p{
```

```
        font-size: size(size-body);
        margin-bottom: 1rem;
        color: var(--text-color-regular);
    }
}

h1{
    grid-area: heading;
    font-family: $font2;
    font-size: size(size-h1);
    color: var(--text-color-heading);
    margin-bottom: 1rem;

}
```

```
#writer-link{
    grid-area: author;

    font-size: size(size-body);
    color: var(--text-color-regular);

    margin-bottom: 2rem;

    a{
        font-weight: 600;
        text-decoration: underline;
        color: var(--text-color-heading);
    }
}
```

```
#related-categories-container{
    grid-area: related-categories;
    position: relative;

}
```

```
#stars{
    grid-area: stars;
    display: grid;
    grid-auto-flow: column;
    grid-auto-columns: max-content;
    align-items: center;

    margin-bottom: 3rem;

    i{
        color: var(--star-orange);
    }
}
```

```
p{
    font-size: size(size-body);
    margin-left: 1rem;
    color: var(--text-color-regular);

    span{
        color: var(--text-color-heading);
    }
}

#book-related-categories-section{
    grid-area: about-book;
}

.article-info-container{
    grid-auto-columns: 220px;
    gap: 20px;
}

#related-books{
    margin-top: 5rem;
}

.dark-mode{
    #div-info-section{
        border-bottom: 1px solid var(--gray-text-color);
    }
}

#div-info-section{
    padding-bottom: 1rem;
    border-bottom: 1px solid var(--category-outline);

    p{
        color: var(--text-color-heading);
    }
}

#review-section{

    margin-top: 8rem;
    margin-bottom: 5rem;

    h2{
        font-family: $font2;
```

```
        color: var(--text-color-heading);
        margin-bottom: 1.5rem;
    }
```

```
#review-stars-container{
    font-size: size(size-body);
    i{
        color: var(--star-orange);
    }
}
```

```
display: grid;
grid-auto-flow: column;
grid-auto-columns: min-content;
align-items: center;
```

```
margin-bottom: 0.5rem;
```

```
p{
    margin-left: 1rem;
    color: var(--gray-text-color);
    font-weight: 600;
}
}
```

```
#review-info{
    display: grid;
    grid-auto-flow: column;
    grid-auto-columns: max-content;
```

```
p{
    margin-right: 1rem;
    font-size: size(size-small);
    font-weight: 500;
    color: var(--gray-text-color);
}
}
```

```
#comments-container-div{
    padding-top: 5rem;
}
```

```
.article-comment{
    display: grid;
    grid-template-columns: 300px 1fr;
    grid-template-rows: auto;
    grid-template-areas: "user-info comment";
```

```
@include md{
```

```
grid-template-columns: 1fr;
grid-template-rows: auto;
grid-template-areas: "user-info"
                     "comment";
}

padding-top: 3rem;
padding-bottom: 3rem;

border-bottom: 1px solid var(--category-outline);
}

.dark-mode{
    .article-comment{
        border-bottom: 1px solid var(--gray-text-color);
    }
}

.comment-user-info-container{
    grid-area: user-info;

    @include md{
        margin-bottom: 1rem;
    }
}

.avatar-and-name-container{
    display: grid;
    grid-auto-flow: column;
    grid-auto-columns: max-content;
    grid-template-rows: auto;
    align-items: center;

    img{
        height: 70px;
        border-radius: 50%;
        margin-right: 0.3rem;
    }
}

p{
    font-size: size(size-body);
    font-weight: 600;
    color: var(--text-color-heading);
}

.comments-stars{
    margin-top: 0.5rem;
    display: grid;
    grid-auto-flow: column;
    grid-auto-columns: max-content;
```

```
    align-items: center;

    i{
        color: var(--star-orange);
    }

    p{
        margin-left: 0.3rem;
        color: var(--gray-text-color);
    }
}

.comment-text{
    //padding-top: 1rem;
    p{
        color: var(--text-color-regular);
        margin-bottom: 1rem;
        white-space: pre-wrap;
    }
}

.comment-pictures-container{
    display: flex;
    flex-wrap: wrap;

    img{
        height: 250px;
        margin-right: 1rem;
        margin-bottom: 1rem;
    }
}

#star-rating-form{
    margin: 0 auto;

    margin-bottom: 5rem;
    width: 800px;
    height: auto;
    padding: 0 0.5rem;

    @include lg-940{
        width: 100%;
    }

    h2{
        text-align: center;
        font-size: 2rem;
        margin-bottom: 3rem;
    }
}
```

```
        color: var(--text-color-heading);  
    }
```

```
.server-messages{  
    text-align: center;  
    padding-top: 1rem;  
}  
}
```

```
.star-raiting-container{  
    display: flex;  
    flex-direction: row;  
    //padding-left: 1rem;  
    justify-content: center;  
}
```

```
div{  
    margin-right: 1rem;
```

```
    border-radius: 3px;  
    background-color: var(--gray-bg-color);  
    outline: 1px solid var(--category-outline);  
    font-size: 1.75rem;  
    padding: 0.3rem;  
    color: var(--star-orange);
```

```
&:hover{  
    cursor: pointer;  
    outline: 2px solid var(--category-outline-hover);  
    background-color: var(--gray-bg-color-hover);
```

```
}  
}
```

```
.dark-mode .star-raiting-container div{  
    &:hover{  
        outline: 2px solid var(--background-color-hover-const)  
    }  
}
```

```
#comment-container{  
    margin-bottom: 1rem;  
}
```

```
#form-comment{  
    .p-title{  
        font-size: 1.25rem;  
        margin-top: 1rem;  
        margin-bottom: 0.3rem;
```

```
        font-weight: 500;
        color: var(--text-color-heading);
    }
```

```
#comment-images-container{
    display: flex;
    flex-direction: column;
    flex-wrap: wrap;
    input{
        margin-bottom: 0.3rem;
    }
}
```

```
//----- BOOK.PHP END -----
//-----
```

```
// ----- CHOOSING CATEGORIES -----
// -----
```

```
.blur{
    animation: changing-blur 0.4s forwards;
}
```

```
.un-blur{
    animation: un-blur 0.4s forwards;
}
```

```
.choosing-categories-section{
    position: fixed;
    top: 0;
    width: 100%;
    height: 100%;
    z-index: 20;
```

```
    display: grid;
    justify-content: center;
    align-items: center;
}
```

```
.choosing-categories-container{
    background-color: var(--background-color);
    padding: 0.5rem;
    width: 1000px;
    max-height: 500px;
    overflow-y: scroll;
    z-index: 23 ;
```

```
    justify-content: center;
```

```
    align-items: center;
    display: grid;
    grid-template-rows:max-content max-content max-content max-content;
    grid-template-columns: 280px 280px 280px;

    grid-template-areas: "title title title";
    gap: 20px;

@include xl{
    width: 700px;

    grid-template-rows:max-content max-content max-content max-
content max-content max-content;
    grid-template-columns: 280px 280px;
    grid-template-areas: "title title";
}

@include md{
    width: 320px;
    grid-template-rows:max-content max-content max-content max-
content max-content max-content max-content max-content max-
content max-content;
    grid-template-columns: 280px;
    grid-template-areas: "title";
}

h2{
    text-align: center;
    grid-area: title;
    color: var(--text-color-heading);
}

article{
    position: relative;
    &:hover{
        padding: 10px;
        border-width: 1px;
        border-color: var(--category-outline);
    }
}

.category-checkbox-choose{
    height: unset;
    position: absolute;
    top: -1px;
    bottom: -1px;
    left: -1px;
    right: -1px;
```

```
    z-index: 25;
    appearance: unset;
```

```
    &:checked{
        border: 4px solid var(--text-color-green);
    }
}
```

```
// ----- CHOOSING CATEGORIES END -----
// -----
```

```
// ----- WEBSITE EXPERIENCE -----
// -----
```

```
#survey-website-experience-section{
    position: relative;
    margin-bottom: 5rem;
```

```
h2{
    font-family: "Source Serif Pro";
    color: var(--text-color-heading);
    margin-bottom: 2rem;
}
```

```
form{
    display: flex;
    justify-content: space-between;
```

```
    flex-wrap: wrap;
```

```
    overflow: hidden;
```

```
@include sm{
    flex-direction: column;
}
}
```

```
article{
    border: 1px solid var(--border-book-shadow);
    background-color: var(--gray-bg-color);
    padding: 0.5rem 0.8rem;
    border-radius: 5px;
    background-color: var(--gray-bg-color);
```

```
@include sm{
    display: flex;
    align-items: center;
```

```
        margin-bottom: 1rem;

    p{
        margin-left: 2rem;
    }

}

&:hover{
    cursor: pointer;
    box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
    background-color: var(--gray-bg-color-hover);
}

img{
    height: 80px;
}

p{
    text-align: center;
    font-weight: 600;
    font-size: size(size-body);
    color: var(--text-color-heading);
}

}

// ----- WEBSITE EXPERIENCE END -----
```

```
//----- SHOP.PHP -----
```

```
#section-container{
    min-height: calc(100vh - 50px);
    display: grid;
    grid-template-columns: 300px auto;
    grid-template-rows: auto;
    grid-template-areas: "filter-sort article-section";

@include lg-940{
    grid-template-columns: auto;
    grid-template-rows: auto auto;
    grid-template-areas: "filter-sort" "article-section";
}

.book-category-input{
    color: var(--text-color-regular);
    margin-bottom: 1rem;
    p{
```

```
i{
    margin-right: 0.5rem;
}

border: 1px solid var(--border-book-shadow);
background-color: var(--gray-bg-color);
padding: 0.3rem;
border-radius: 5px;

&:hover{
    cursor: pointer;
}
}

font-size: size(size-body);

.book-category-li{

padding: 0 1.5rem;
display: grid;
grid-template-columns: 20px 1fr;
gap: 5px;
grid-template-rows: 1fr;
align-items: center;

label{
    &:hover{
        cursor: pointer;
    }
}
}

}

#filter-sort-section{
height: fit-content;
//position: sticky;
top: 50px;
border: 1px solid var(--category-outline);
padding-bottom: 1rem;
background-color: var(--gray-bg-color);

@include lg-940{
    justify-self: center;
    width: 300px;
    position: relative;
    top: unset;
}
```

```
        border: unset;
        border-bottom: 1px solid var(--category-outline);

    }
}
```

```
#filter-sort-section{
    padding-top: 3rem;
    padding-left: 2rem;
    padding-right: 2rem;
}
```

```
#input-search-container{
    width: 100%;
    position: relative;
```

```
i{
    position: absolute;

    top: 10px;
    right: 10px;
    font-size: 22px;
}
}
```

```
#input-search{
    margin-bottom: 1rem;
    height: 40px;
    border-radius: 50px;
    padding: 0 2rem;
    padding-right: 40px;
    width: 100%;
```

```
    font-size: size(size-body);
```

```
    border: 1px solid var(--gray-text-color);
```

```
&:focus {
    border-color: transparent;
    outline: 2px solid var(--background-color-header-const);
    box-shadow: 0 2px 10px rgb(0 0 0 / 6%);
}
```

```
&:focus-visible{
    outline: 2px solid var(--background-color-header-const);
}
```

```
}
```

```
#heading-price{
    margin-bottom: 0.5rem;
    font-size: size(size-body);
    color: var(--text-color-heading);
}
```

```
#price-container{
    display: grid;
    grid-auto-flow: column;
    //grid-template-columns: 1fr 1fr;
    justify-content: space-between;
```

```
    margin-bottom: 1rem;
```

```
input{
    width: 100px;
    height: 30px;
    font-size: 20px;
    padding: 0 5px;
}
}
```

```
#heading-categories{
    color: var(--text-color-heading);
    margin-bottom: 0.5rem;
    font-size: size(size-body);
}
```

```
#heading-sort{
    color: var(--text-color-heading);
    margin-bottom: 0.5rem;
}
#sort{
    width: 100%;
    height: 30px;
    font-size: 20px;
}
```

```
.dark-mode{
    #input-search{
        &:focus{
            outline-width: 2px;
            outline-color: var(--gray-text-color);
        }
    }
}
```

```
#price-min{
```

```
    &:focus{
        outline-width: 2px;
        outline-color: var(--gray-text-color);
    }
}
```

```
#price-max{
    &:focus{
        outline-width: 2px;
        outline-color: var(--gray-text-color);
    }
}
```

```
#sort{
    &:focus{
        outline-width: 2px;
        outline-color: var(--gray-text-color);
    }
}
```

```
#section-articles{
    padding: 3rem 0 3rem 5rem;
}

}
```

```
#articles-container-ul{
    display: flex;
    flex-wrap: wrap;

    @include md{
        justify-content: center;
    }
}
```

```
.li-tag-article-container{
    width: 180px;
    height: 400px;

    article{
        height: 100%;
    }
    margin-right: 40px;
    margin-bottom: 40px;
    // height: max-content;
}
```

```
.dark-mode{
  .pagination{
    li{
      &:hover{
        outline: 2px solid var(--background-color-hover-const);
      }
    }
  }
}

  .active{
    background-color: var(--background-color-header-const) !important;
  }
}

.pagination{
  display: flex;
  flex-wrap: wrap;
}

  .active{
    background-color: var(--bg-button-hover) !important;
  }

  li{
    border-radius: 3px;
    margin-right: 0.8rem;
    margin-bottom: 1rem;
    width: fit-content;
    height: fit-content;
  }

  background-color: var(--gray-bg-color);

  outline: 1px solid var(--category-outline);

  &:hover{
    outline: 2px solid var(--category-outline-hover);
    background-color: var(--gray-bg-color-hover);
  }

  font-size: size(size-body);
  a{
    color: var(--text-color-regular);

    &:hover{
      color: var(--text-color-heading);
    }
  }

  display: inline-block;
```

```
        height: 40px;
        width: 40px;
        display: grid;
        justify-content: center;
        align-items: center;
    }
}

}
```

```
//----- SHOP.PHP END -----
//-----
```

```
//----- CHECKOUT.PHP -----
//-----
```

```
#cart-items{
```

```
    padding-top: 3rem;
    //min-height: calc(100vh - 50px);
```

```
#table-container{
    overflow-x: scroll;
    margin-bottom: 2rem;
}
```

```
table{
    border-collapse: collapse;
    border-spacing: 0;
```

```
    width: 100%;
```

```
@include xl{
    width: 1200px;
}
thead{
    height: 2rem ;
    background-color: var(--background-color-header-const);
    font-weight: 600;
    font-size: size(size-body);
    color: var(--text-color-heading-const);
    width: 100%;
```

```
td{
    padding-left: 0.5rem;
```

```
        }

    }

.td-image-checkout{
    width: 200px ;

}

tbody{

    .td-remove{
        i{
            color: var(--text-color-heading);
            border: 1px solid var(--category-outline);
            background-color: var(--remove-button-bg);
            padding: 0.5rem;

            &:hover{
                cursor: pointer;
                background-color: var(--remove-button-bg-hover);
                box-shadow: 0 2px 2px rgb(0 0 0 / 10%);
            }
        }
    }

tr{
    &:hover{
        background-color: var(--table-tr-hover) !important;
    }
}

.td-image{
    display: grid;
    justify-content: center;
    padding-top: 0.5rem;
    padding-bottom: 0.5rem;

    img{
        height: 120px;
    }
}

.td-unit-price, .td-price{
    p{
        color: var(--text-color-heading);
        font-size: size(size-body);
```

```
        font-weight: 600;
```

```
    }  
}
```

```
.td-title{  
    position: relative;  
    div{  
        position: absolute;  
        top: 0;  
        bottom: 0;  
        left: 0;  
        right: 0;  
        padding: 0.5rem;  
        color: var(--text-color-regular);  
        font-size: size(size-body);  
    }  
}
```

```
.td-quantity{  
    div{  
        display: grid;  
        justify-content: center;  
        input{  
            width: 100px;  
            text-align: center;  
            height: 30px;  
            font-size: size(size-body);  
        }  
    }  
}
```

```
.td-title{  
    width: 400px;  
}
```

```
.td-unit-price{  
    width: 150px;  
    text-align: center;  
}
```

```
.td-quantity{  
    width: 150px;  
    text-align: center;  
}
```

```
.td-price{
```

```

        width: 150px;
        text-align: center;
    }
    .td-remove{
        width: 150px;
        text-align: center;
    }
}
}

#empty{
    width: 100%;
    padding-top: 2rem;
    padding-bottom: 2rem;
    background-color: rgba(255, 0, 0, 0.217);

    text-align: center;
    font-size: size(size-h3);
    color: var(--text-color-heading);
    font-weight: 600;
}

}

#total-price-container{
    text-align: end;
    padding-right: 1rem;
    font-size: size(size-h3);
    font-weight: 600;
    color: var(--text-color-heading);
}

#checkout-address-container{
    grid-template-areas:
    "address-line-title address-line-title address-line-title"
    "address-line-input address-line-input address-line-input"
    "address-line-error address-line-error address-line-error"

    "number-title . delivery-title"
    "number-input . delivery-input"
    "number-error . delivery-error"

    "city-title . state-title"
    "city-input . state-input"
    "city-error . state-error"

    "zip-code-title . country-title"
    "zip-code-input . country-input"
    "zip-code-error . country-error";
}

@include xmd{

```

```
grid-template-rows: repeat(21,auto);
grid-template-areas:
"address-line-title"
"address-line-input"
"address-line-error"

"delivery-title"
"delivery-input"
"delivery-error"

"number-title"
"number-input"
"number-error"

"city-title"
"city-input"
"city-error"

"state-title"
"state-input"
"state-error"

"zip-code-title"
"zip-code-input"
"zip-code-error"

"country-title"
"country-input"
"country-error";

}

.delivery-title{
    grid-area: delivery-title;
}
#delivery-input{
    grid-area: delivery-input;
}
#delivery-error{
    grid-area: delivery-error;
}

}

// #form{
//     display: grid;
//     justify-content: center;
//     padding-top: 5rem;
//     padding-bottom: 5rem;
```

```
//      label{  
//          display: block;  
//      }  
  
//      .radio-container{  
//          label{  
//              display: inline-block;  
//          }  
//      }  
  
//      .error-msg{  
//          display: none;  
//          color: rgb(255, 31, 31);  
//          margin-top: 0.3rem;  
//      }  
  
//      form{  
//          padding-left: 1rem;  
//          padding-right: 1rem;  
//          padding-bottom: 1rem;  
//          border: 1px solid var(--category-outline);  
//          box-shadow: 0 2px 2px rgb(0 0 0 / 10%);  
//          background-color: var(--gray-bg-color);  
//          border-radius: 3px;  
//      }  
  
//      input[type=text], input[type=email]{  
//          width: 300px;  
//          height: 30px;  
//          font-size: size(size-body);  
//          font-weight: 600;  
//      }  
  
//      .form-heading{  
//          font-size: size(size-body);  
//          font-weight: 600;  
//          margin-top: 1rem;  
//          margin-bottom: 0.3rem;  
//          color: var(--text-color-heading);  
//      }  
  
//      #agree-radio-container{  
//          margin-top: 1rem;  
//      }  
  
//      label{  
//          color: var(--text-color-regular);  
//      }  
// }
```

```
// #button-container{
//   width: 100%;
//   display: grid;
//   justify-content: center;
//   margin-top: 1rem;

//   button{
//     padding: 0.2rem;
//     font-size: size(size-body);
//     background-color: var(--background-color-header-const);
//     font-weight: 600;
//     color: var(--text-color-heading-const);
//     border: unset;

//     &:hover{
//       cursor: pointer;
//       background-color: var(--background-color-hover-const);
//     }
//   }
// }

//----- CHECKOUT.PHP END -----
//-----
```

```
// ----- AUTHOR.HTML -----
//-----
```

```
#author-section{
  min-height: 100vh;
  margin-bottom: 3rem;
  max-height: fit-content;

  display: flex;
  //justify-content: center;
  align-items: center;
  flex-direction: column;
```

```
h1{
  color: var(--text-color-heading);
  margin: 4rem 0;
  font-size: size(size-big);
}
```

```
h2{
  margin-bottom: 0.3rem;
  color: var(--text-color-heading);
  font-size: size(size-h3);
}
```

```
p{
```

```
    color: var(--text-color-regular);
    margin-bottom: 1rem;
    font-size: size(size-body);
}
```

```
#img-container{
    width: 250px;
    margin-bottom: 2rem;
}
```

```
img{
    width: 100%;
}
}
```

```
#four-o-four{
    min-height: calc(100vh - 50px);
    display: grid;
    justify-content: center;
    align-items: center;
```

```
h1{
    font-size: 15rem;
    color: var(--background-color-header-const);
}

}
```