Факултет: ФИТА

Специалност: „Софтуерни и интернет и технологии“

**WEB програмиране – проект**

Тема:

**Проектиране и реализиране на Web система**

**на php – БИБЛИОТЕКА**

Изготвил: ......................

Явор Константинов Стоянов

Фак. №: 20621620

Специалност: СИТ

II курс 4 група

**СЪДЪЖАНИЕ**

1. Задание
2. Използвани технологии – разработка на библиотека
3. Провеждане на тестови постановки и поддръжка
4. Функционалност на системата
5. Перспективи за развитие
6. Изходен код на системата
7. Източници

**I ЗАДАНИЕ**

Да се проектира и реализира WEB система на php БИБЛИОТЕКА, която да съхранява следната информация:

* Книга – наименование, автор, година на издаване, издателство, жанр, брой заемания, статус (заета/свободна);
* Служител – име, фамилия, позиция, телефон, e-mail;
* Читател – име, фамилия, телефон, e-mail;
* Заемане – книга, читател, служител, дата на заемане, срок за връщане (дата).

Правила:

* Всяка книга може да бъде заета текущо само от един читател
* Всяка книга има един автор и едно издателство

Системата трябва да може да реализира следните функционалности:

* Въвеждане / корекции / изтриване на данни за всяка таблица.
* Търсене/заемане на книги по: автор, ключови думи от наименованието, жанр.
* Справки за:
  + заети книги от читател
  + не върнати книги, подредени по дата на връщане
  + Свободни книги подредени по брой-заемания
  + Най-често заемани книги
  + Топ 5 класация на служители по брой заети книги
  + Топ 5 класация на читатели по брой заети книги

Създаването на базата и таблиците да става чрез код в php-то, а не чрез phpmyadmin-a. Останалата част от функционалността да се сведе до работа с форми.

Р.S. Да не забравите да си нормализирате таблиците и направете приличен дизайн (да не е бяла грозна страница).

**II Използвани технологии – разработка на библиотека**

**1. HTML**

Абревиатурата HTML e съкращение от Hyper Text Markup Language. Това е език за маркиране на хипертекст. HTML е стандартен за използване в Интернет. Правилата за боравенето с него се определят от Международната организация по стандартизация. Описанието на обработвания документ се извършва чрез специални елементи, които се наричат още и HTML тагове. Тези елементи представляват основната градивна единица на уеб страниците. Те са важни, заради тяхната функция да се оформят отделните части от текста на една уеб страница. Такива части могат да бъдат: заглавия, цитати, раздели, хипертекстови препратки и др. Текстовите файлове съдържат съдържание с маркери- инструкции за браузъра, с които по-лесно се показва текста. Основната функция на уеб браузърите е разчитането на документите с HTML езика, след което да ги превърнат в уеб страници. HTML таговете не се показват в браузърите, а се използват, при интерпретирането на съдържанието на една страница.



Фиг 1.1 – Официално лого на HTML

**2. CSS**

CSS (Cascading Style Sheets) е създаден първоначално като инструмент, с чиято помощ се разделя съдържанието от представянето му. В наши дни се използва главно за визуално оформление на HTML страници. Може да се прилага и върху произволен XML документ. CSS представлява удобен начин за форматиране на текст, шрифтове, изображения и други елементи на една уеб страница. Cascading Style Sheets е изключително удобен, когато трябва да се изгражда определен HTML документ.



Фиг. 2.1 – Официално лого на CSS

**3. SQL**

Structured Query Language, SQL е популярен език за програмиране, предназначен за създаване, видоизменяне, извличане и обработване на данни от релационни системи за управление на бази данни.



Фиг 3.1 SQL

**4. PHP**

За свързването на проекта със сървър се използва скриптов език - PHP. Той е проектиран специално за създаването на уеб приложения с динамично съдържание. Скриптовият език с отворен код е един от най-популярните, използван за програмиране в Интернет пространството. Скриптовият език с отворен код се използва основно в уеб средите, като с негова помощ се реализират различни видове услуги. Интерпретирането на скриптовия език с отворен код става, когато се направи заявка от потребител на Интернет за зареждане на определена уеб страница. Тогава резултатът от заявката се връща на уеб браузъра и потребителят може да види само резултата от изпълнението на PHP кода. Този код не може да се види, без да има достъп до съответния файл, в който той е записан.



Фиг 4.1 Лого на PHP

**5. Bootstrap**

Bootstrap е CSS рамка, която е сред най-популярните HTML, CSS и JS рамки. Това е безплатен набор от инструменти за създаване на уебсайтове, който включва типографски елементи, готови уеб форми, бутони, менюта и други уеб елементи.



Фиг 5.1 Официално лого на Bootstrap 5

**6. XAMPP**

XAMPP означава Cross-Platform (X), Apache (A), MySQL (M), PHP (P) и Perl (P). Това е лека дистрибуция на Apache, която е изключително удобна за изграждане на тестов локален уеб сървър. Всички приложения, необходими за работата на тестов локален сървър - уеб сървър (Apache), сървър за бази данни (MySQL) и скриптов език (PHP) - са включени в един единствен инсталационен файл.



Фиг 6.1 Официално лого на XAMPP

**7. phpMyAdmin**

Уеб базиран инструмент за администрация на MySQL. Написан е на PHP.

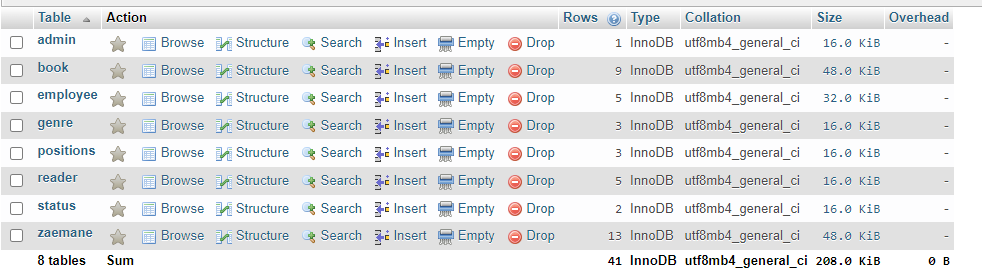
С негова помощ могат да се извършват следните действия: създаване и изтриване на база данни, създаване/изтриване/променяне на таблици, добавяне/изтриване и редактиране на полета, изпълнение на SQL код и менажиране на ключове (keys).

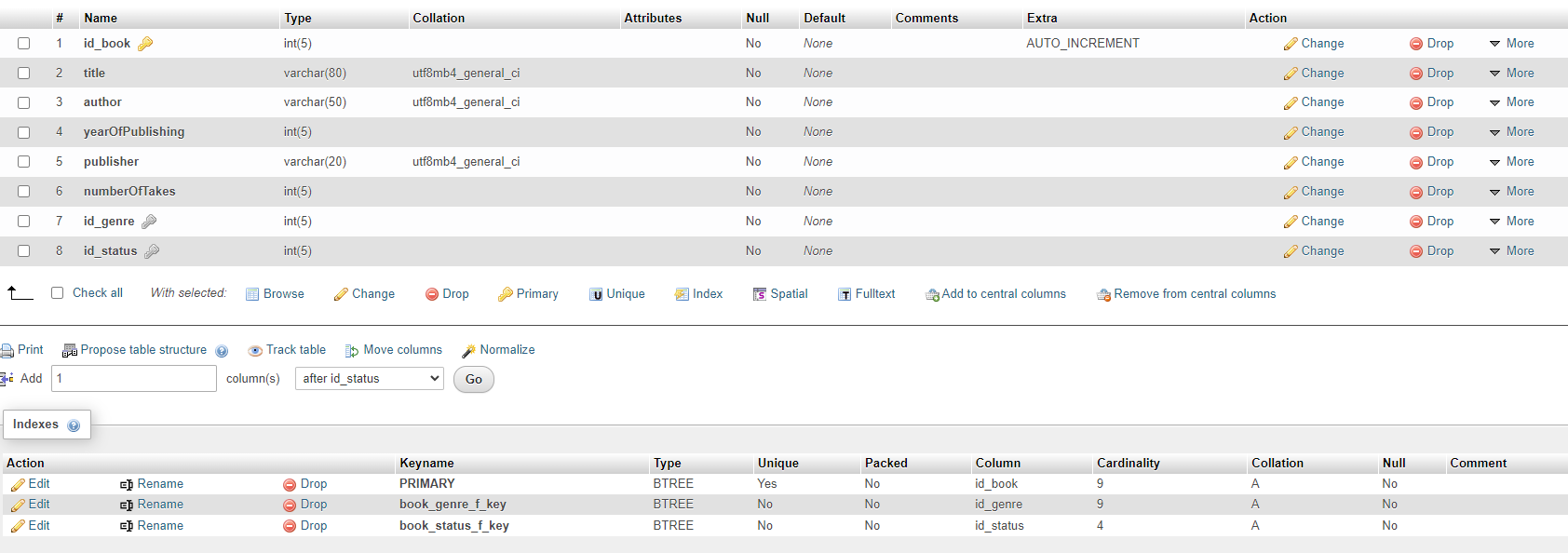


Фиг 7.1 Официално лого на phpMyAdmin

**8. База данни**

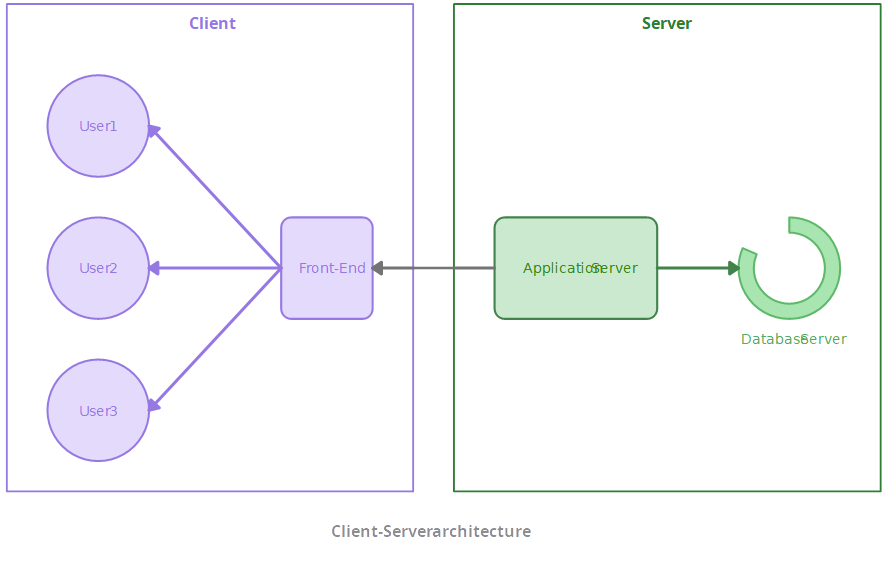
Бази от данни представляват организирана съвкупност от взаимосвързани данни, разположени в един или няколко файла, отразяващи състоянието и измененията в наблюдавани параметри на обекти от предметната област. Те служат като единна информационна среда за приложните програми, чрез които се автоматизира процеса на управление. Данните се организират така, че да има възможност за добавяне на нови и модифициране на съществуващите, а така също и за осъществяване на успешно информационно търсене. Системата използва 1 база от данни с 8 таблици в нея всяка от която отговаря за различна функционалност на системата.

Фиг 8.1 Структура на базата данни на проекта

Фиг 8.2 Примерен изглед на таблица – табл. Книги

**9. Архитектура**

Клиент-сървър архитектура на най-често се използва за създаване на корпоративни бази данни, където информацията не се съхранява само, но и от време на време за обработка на различни методи. Тази база данни е ключов елемент на всяко предприятие информационна система, а сървърът е в основата на тази рамка. Така, сървърът настъпи най-сложните операции на въвеждане, съхранение, обработка и модификация на данни. Когато потребител (клиент) достъп до базата данни (сървър), заявката се обработва: директно достъп до база данни и връща отговор (резултат обработка). В резултат на обработка - мрежа съобщение за успешна операция или на грешка.

****

Фиг 9.1 Графика на архитектурата клиент- сървър

**III Провеждане на тестови постановки и поддръжка**

Преди да бъде качен в интернет, всеки софтуер трябва да бъде тестван и проверен щателно за допуснати грешки , както откъм гледна точка на защитата така и откъм производителност и скорост на работа. Софтуерното тестване е процес на изследване и проучване на софтуер, с цел получаване на информация за качеството на продукта и услугата, която се изпитва. Процесите на софтуерното тестване са неразделна част от софтуерното инженерство и осигуряване на качеството на софтуера.

**1. Тестване**

Системата е тествана с повечето възможни браузъри – Chrome, Mozilla, Opera и Internet Explorer. Тестовете показват отлични резултати по всички изброени параметри.

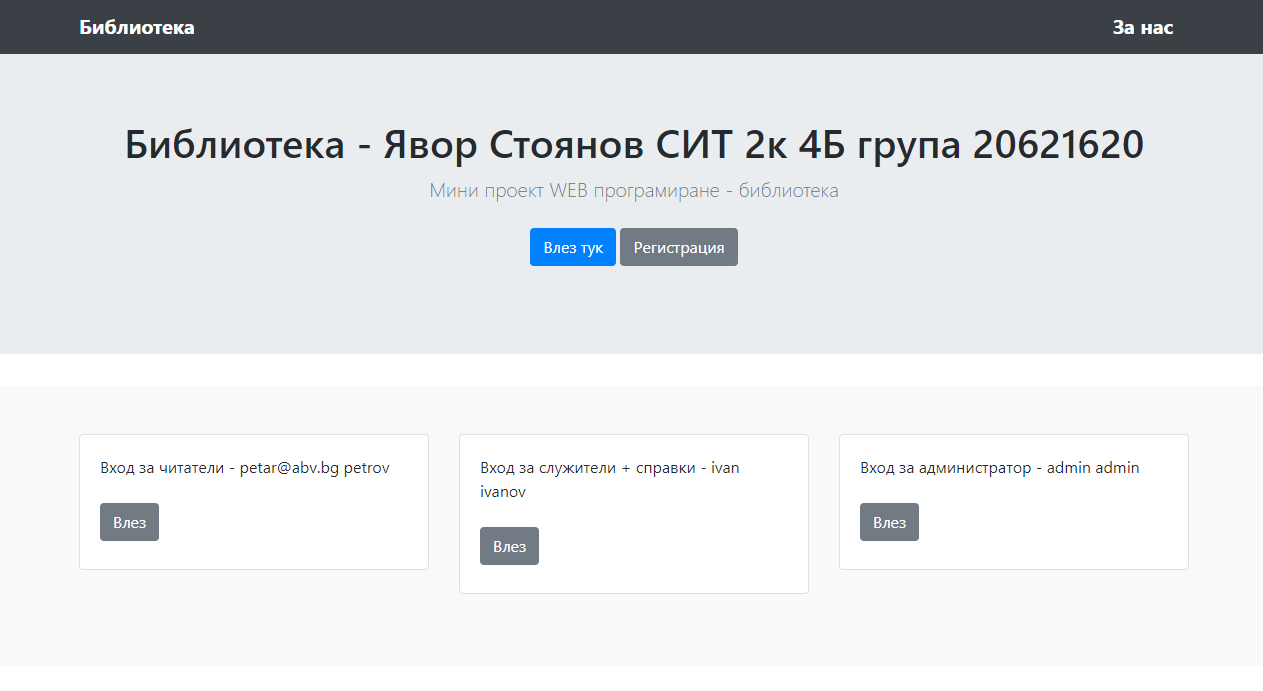
**2. Поддръжка на системата**

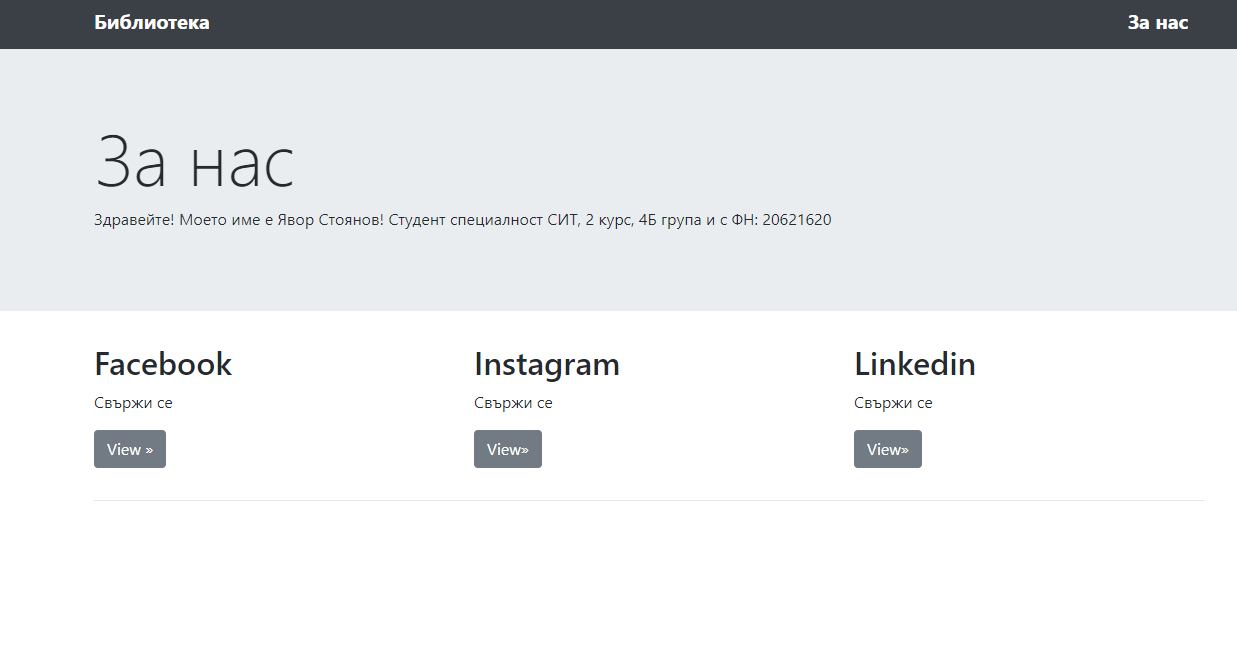
Поддръжката на системата се състои в отстраняване на грешки, пропуснати при тестването преди официалното пускане както и добавяне на нови функции към сайта. За в бъдеще е нужно да се изгради екип от хора, които да следят за правилната работа на сайта и да подобрява функциите й, с цел да го направи по-лесен и полезен за потребителите.



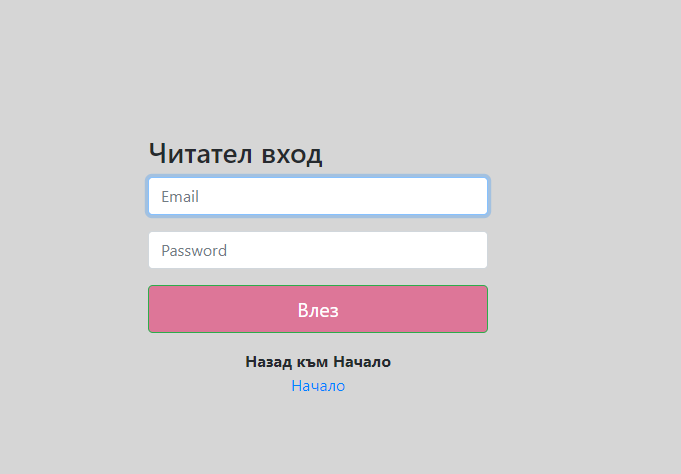
**IV Функционалност на системата**

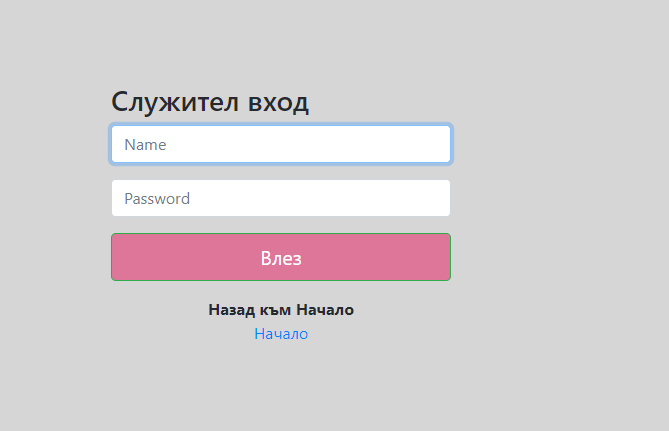
**1. Начална страница**

**2. Страница с контакти – За нас**

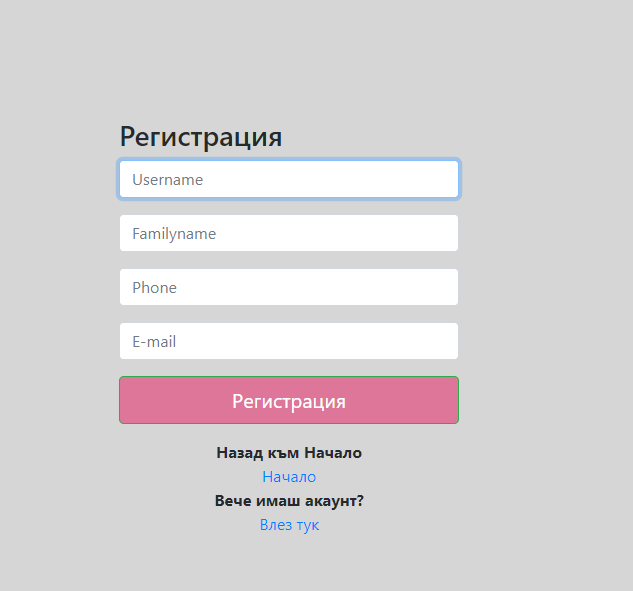


**3. Страница за вход на читател**

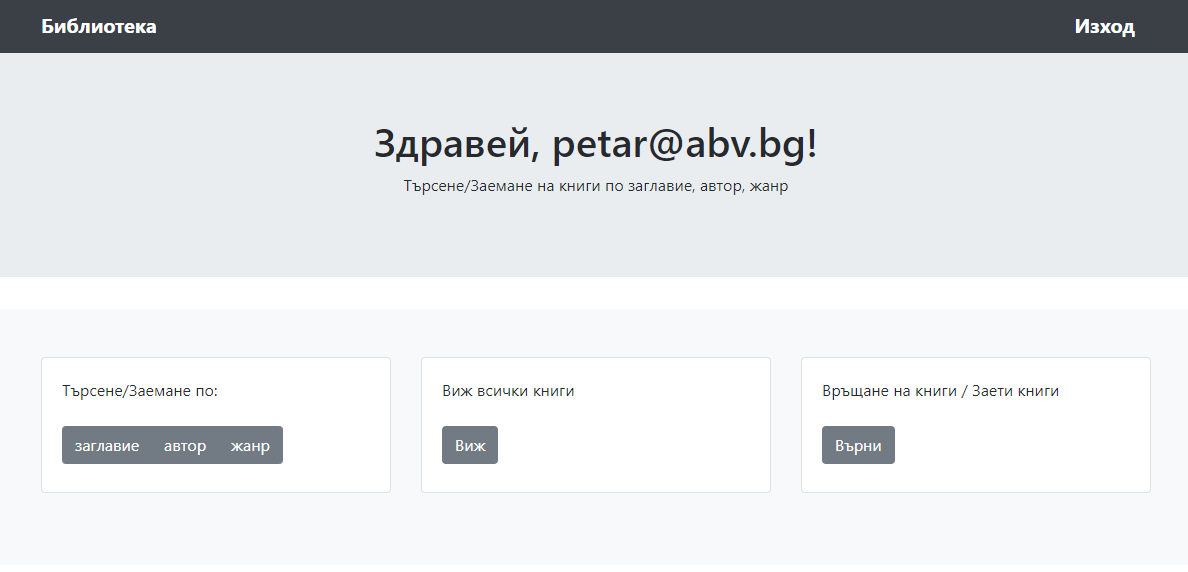
****

**4. Страница за вход на служител**

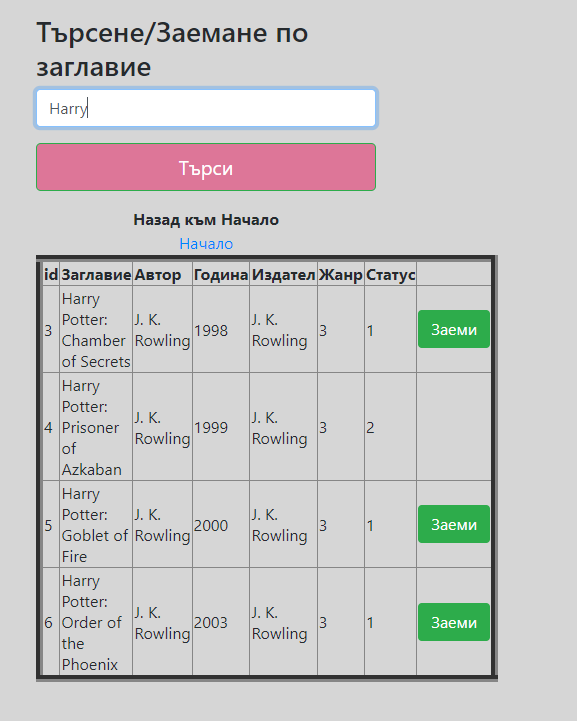
**5. Страница за регистрация на потребител**



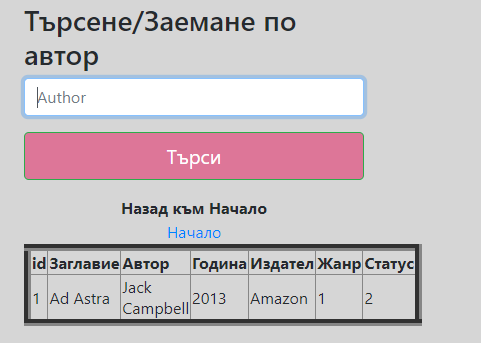
**6. Потребителски панел**



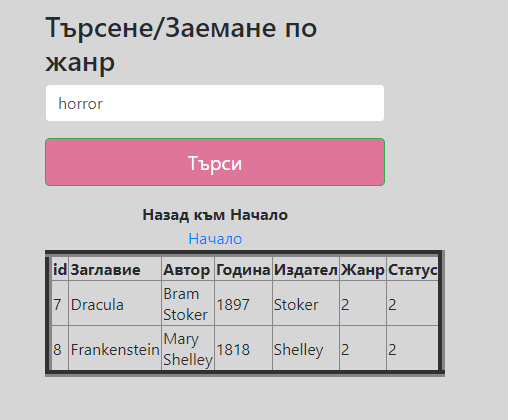
**7. Търсене и заемане на книга по заглавие**



**8. Търсене и заемане на книга по заглавие**



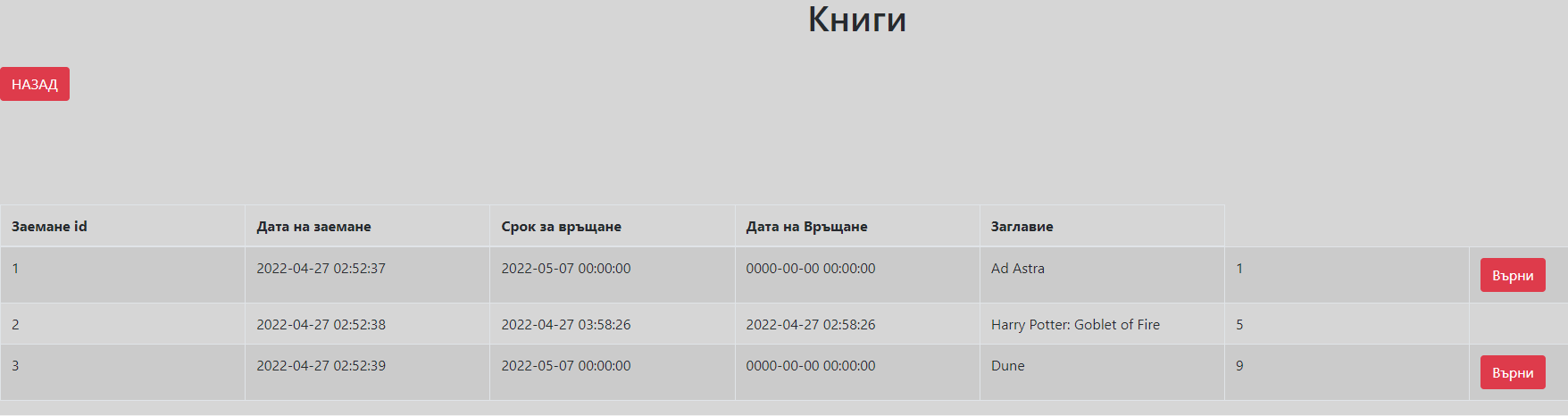
**9. Търсене и заемане на книга по жанр**



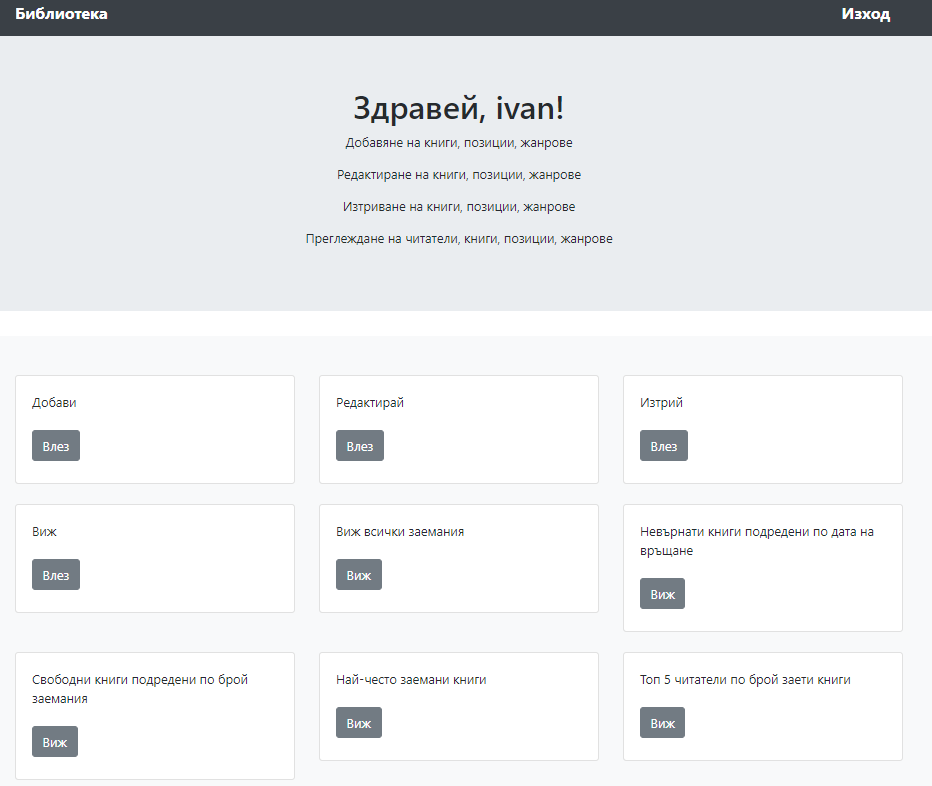
**10. Всички налични книги**



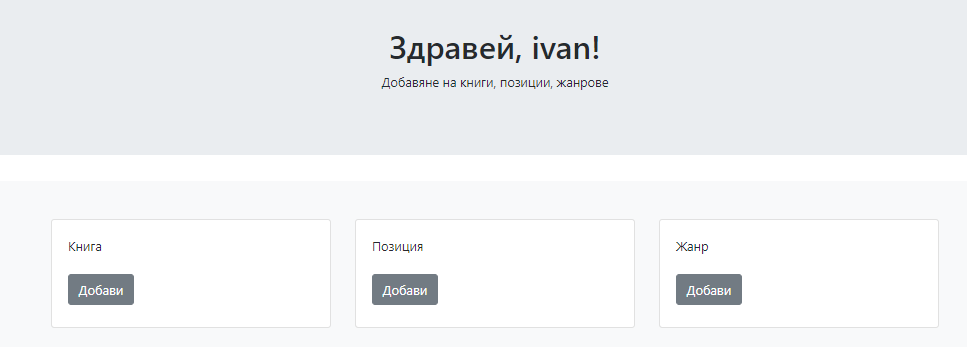
**11. Връщане на книги / Преглеждане на заети книги**



**12. Служителски панел**



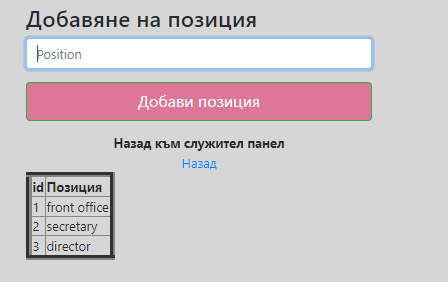
**13. Опции за добавяне от служител**



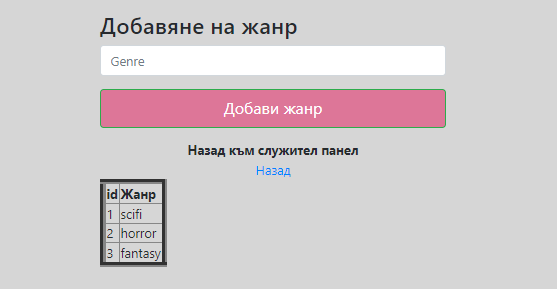
**14. Добавяне на книга от служител**



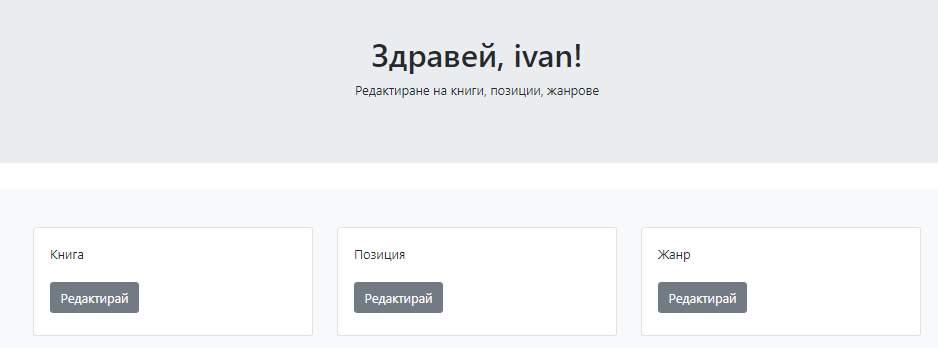
**15. Добавяне на позиция от служител**



**16. Добавяне на жанр от служител**



**17. Опции за редактиране от служител**



**18. Избор книга за редактиране от служител**



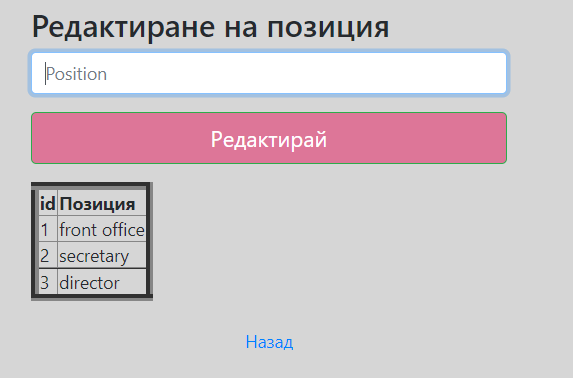
**19. Редактиране на данни на избрана книга**



**20. Избор на позиция за редактиране от служител**



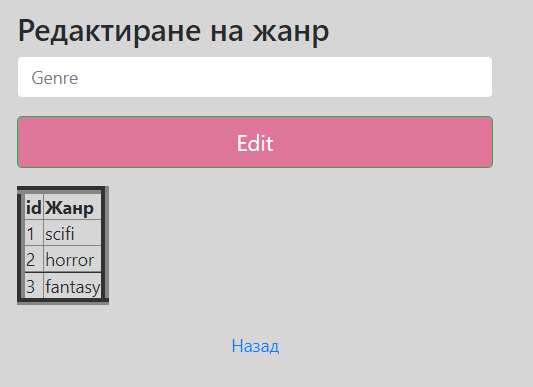
**21. Редактиране на избрана позиция**



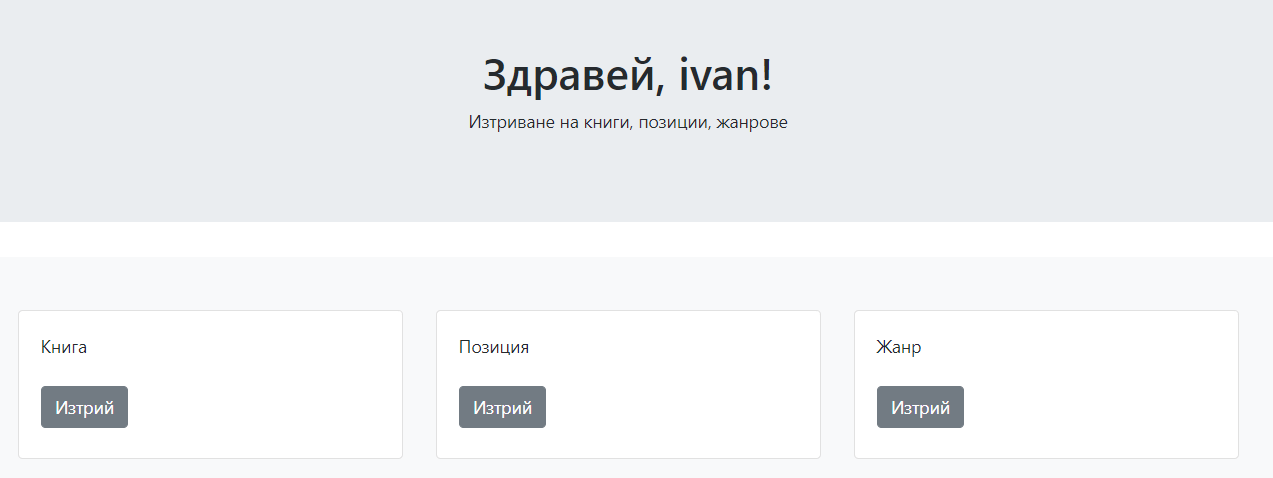
**22. Избор на жанр за редактиране от служител**



**23. Редактиране на избран жанр**



**24. Опции за изтриване от служител**



**25. Избор на книга за изтриване от служител**

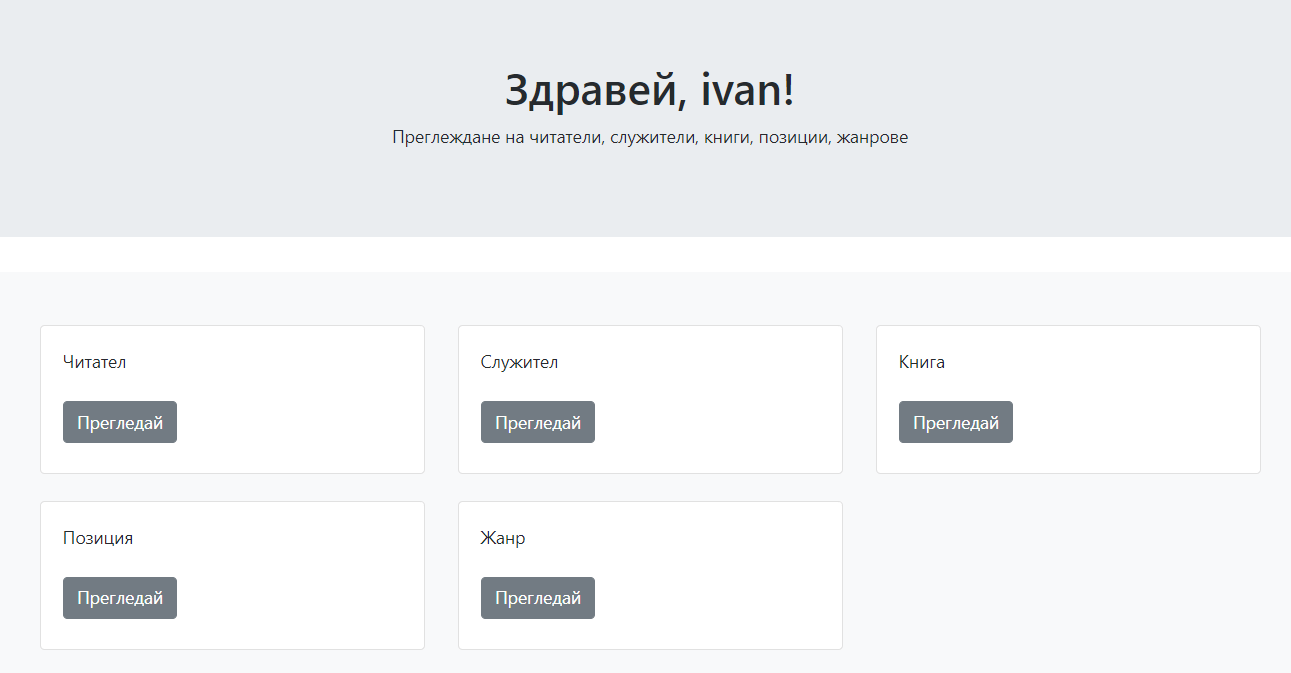


**26. Избор на позиция за изтриване от служител**

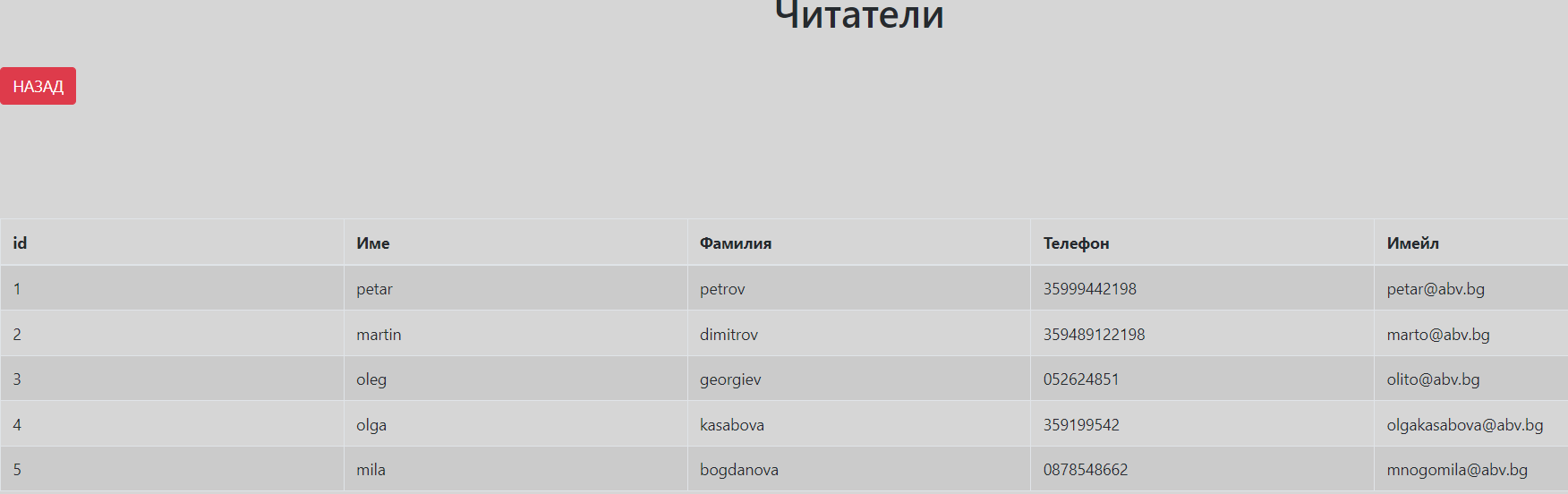


**27. Избор на жанр за изтриване от служител**

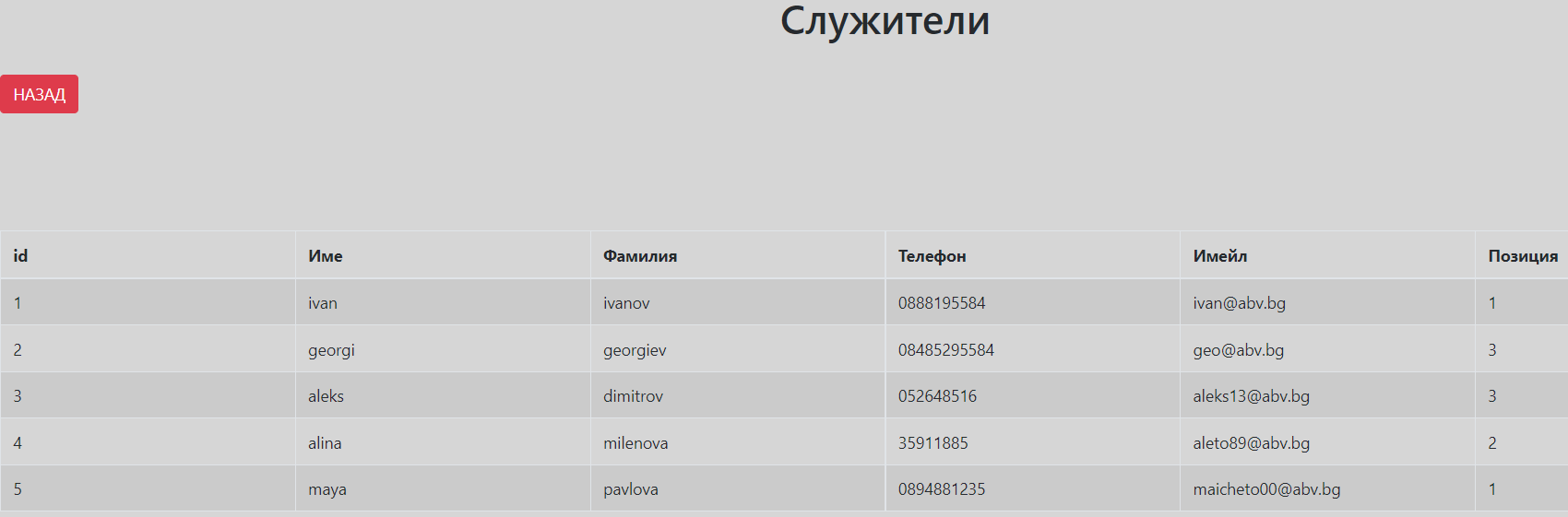


**28. Опции за преглеждане от служител**

**29. Преглед на всички читатели**

****

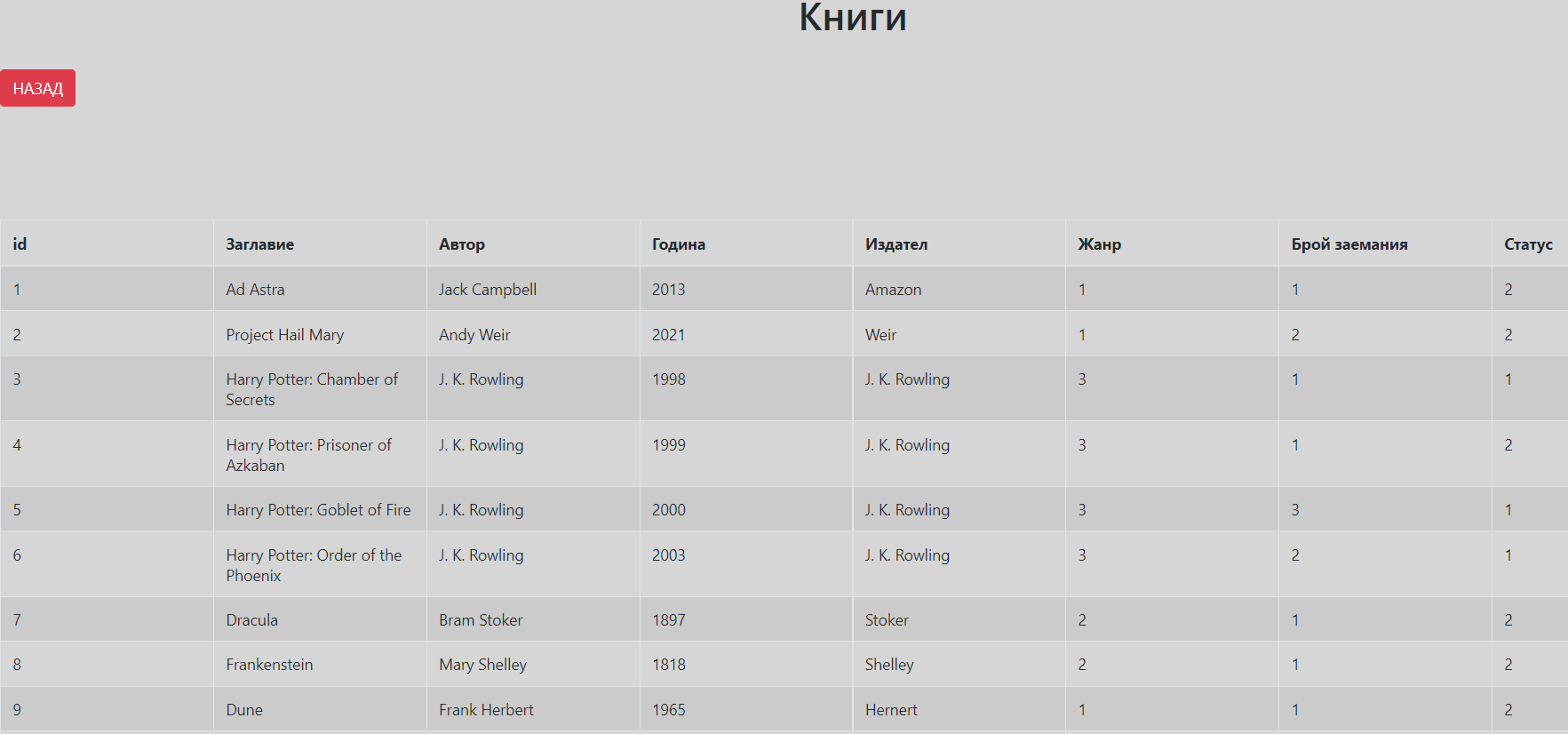
**30. Преглед на всички служители**



**31. Преглед на всички позиции**



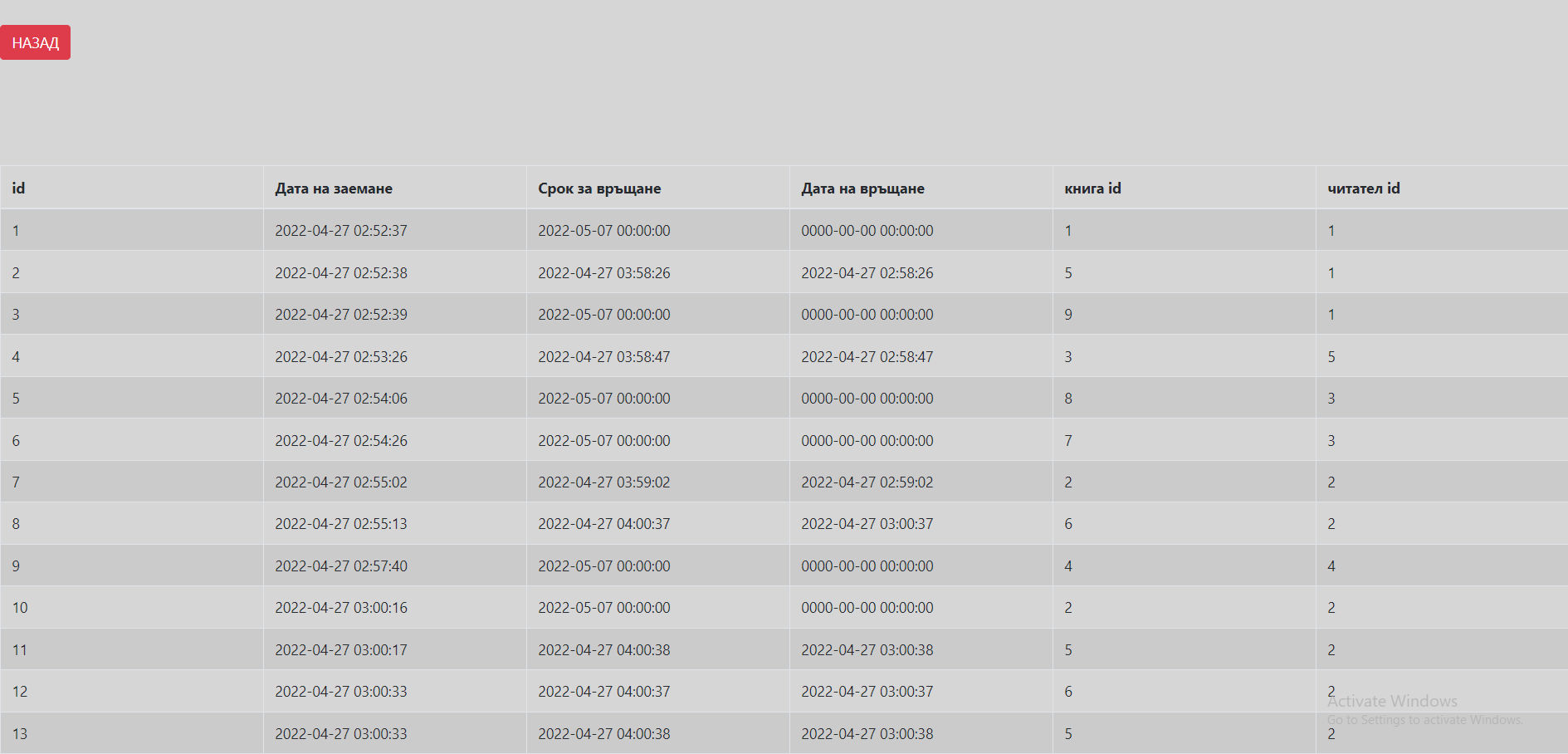
**32. Преглед на всички книги**



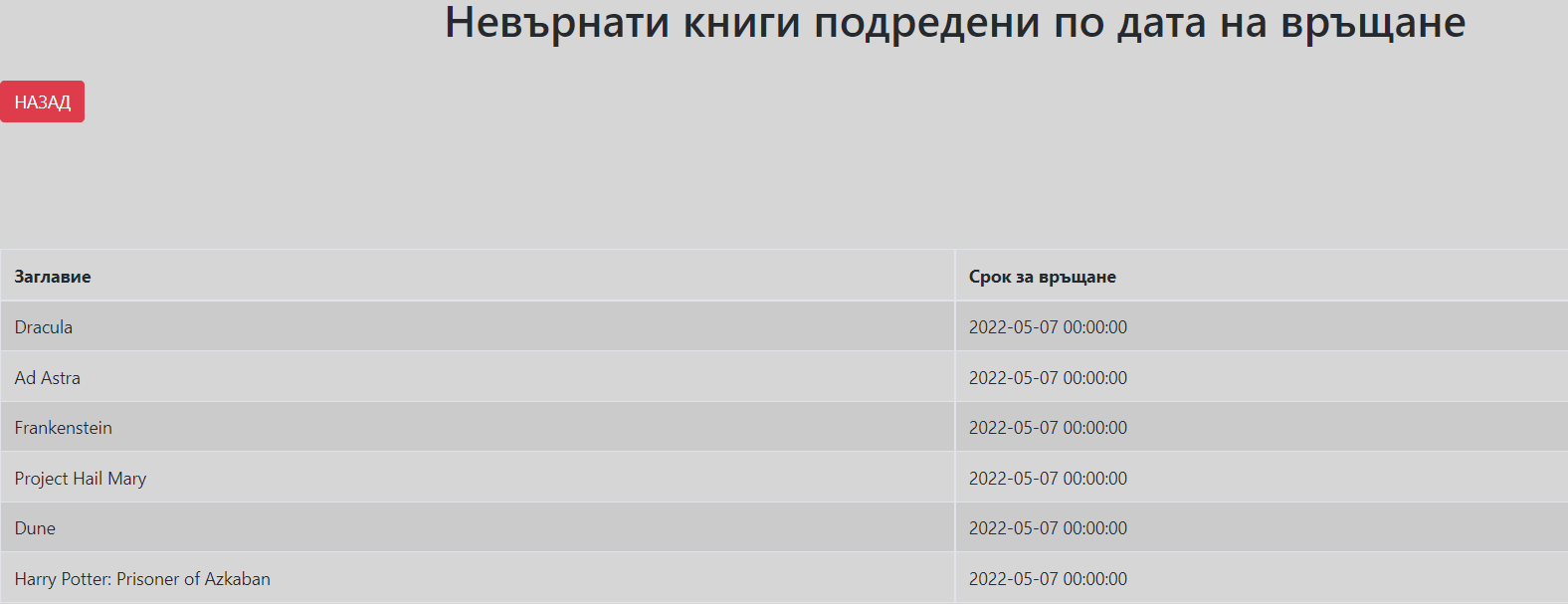
**33. Преглед на всички жанрове**



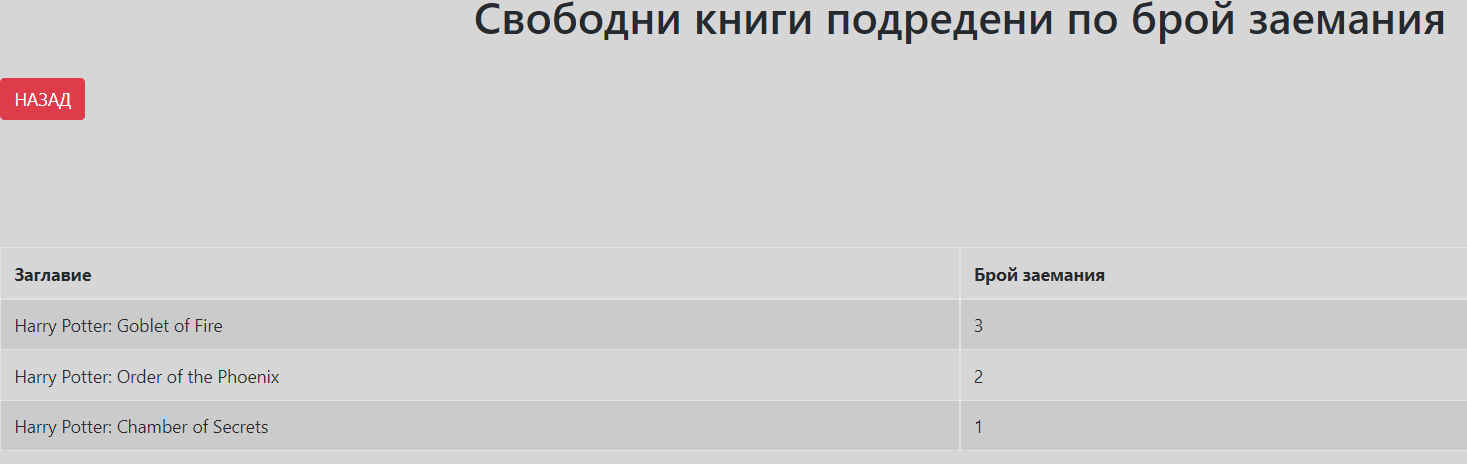
**34. Преглед на всички заемания от читатели**



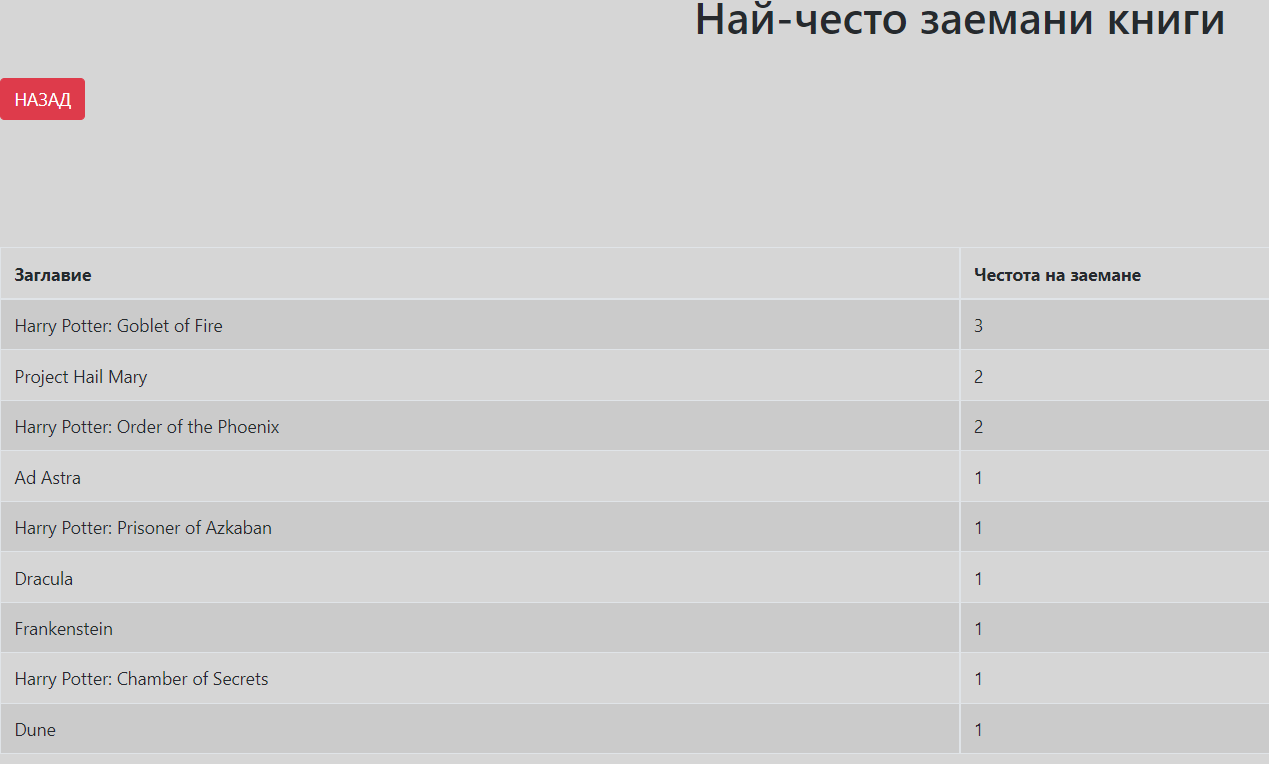
**35. Невърнати книги подредени по дата на връщане**



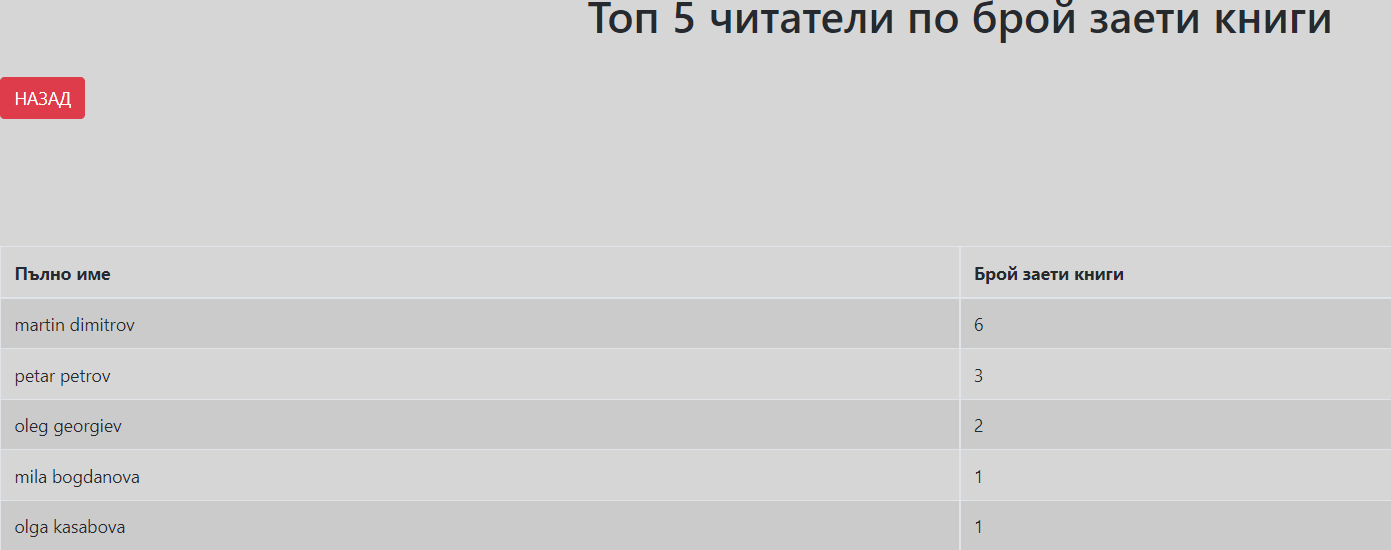
**36. Свободни книги подредени по брой заемания**



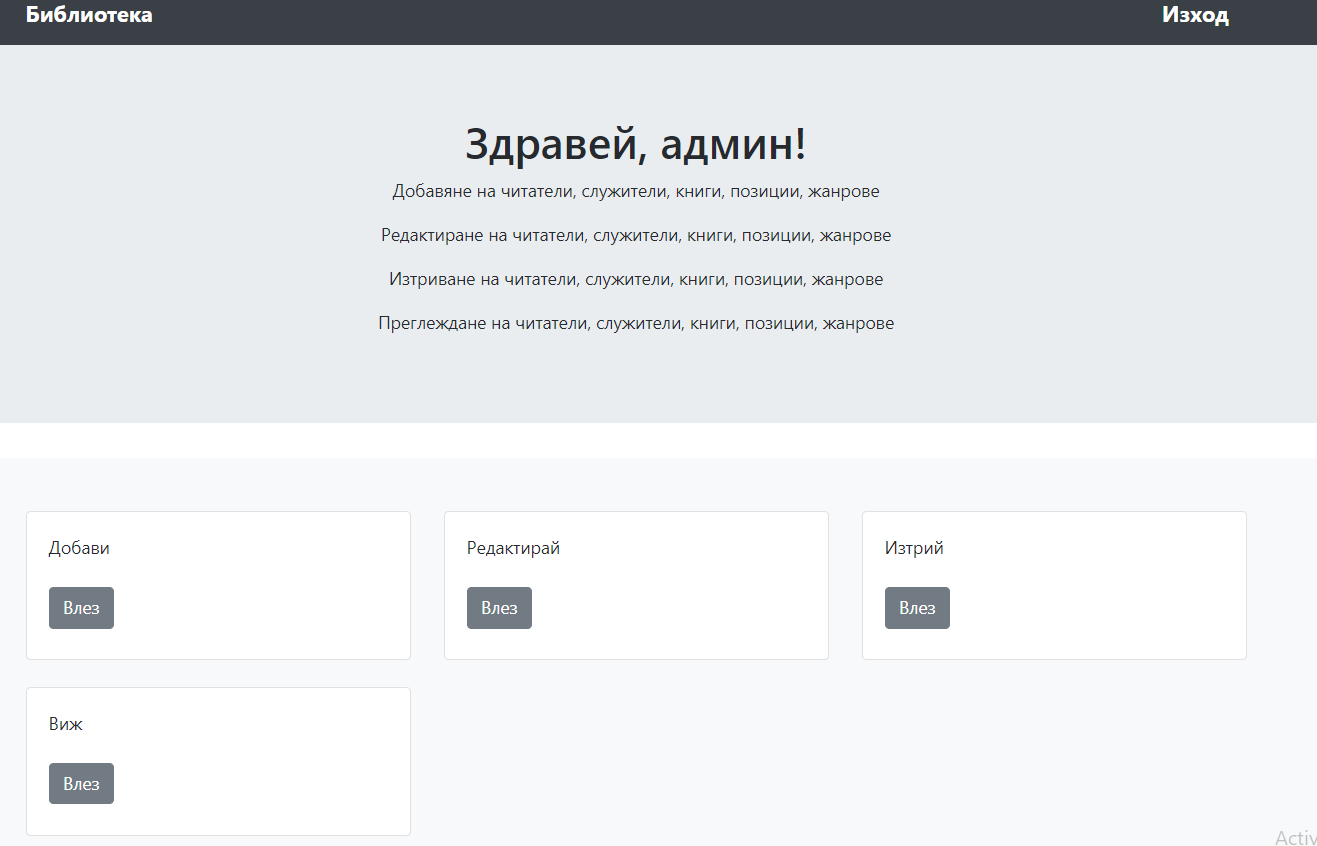
**37. Най- често заемани книги**



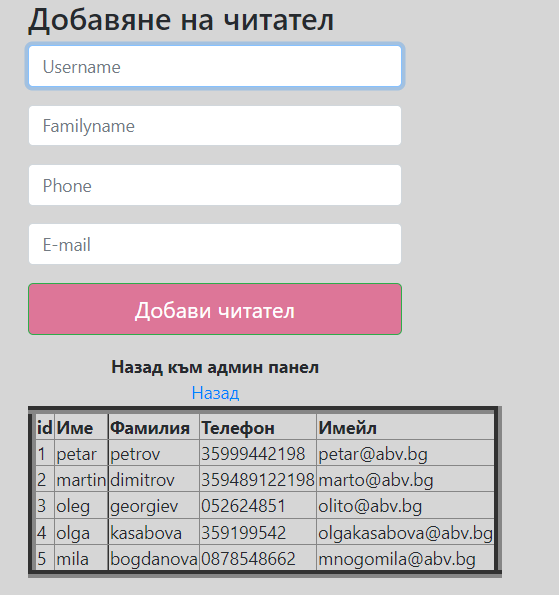
**38. Toп 5 читатели по брой заети книги**



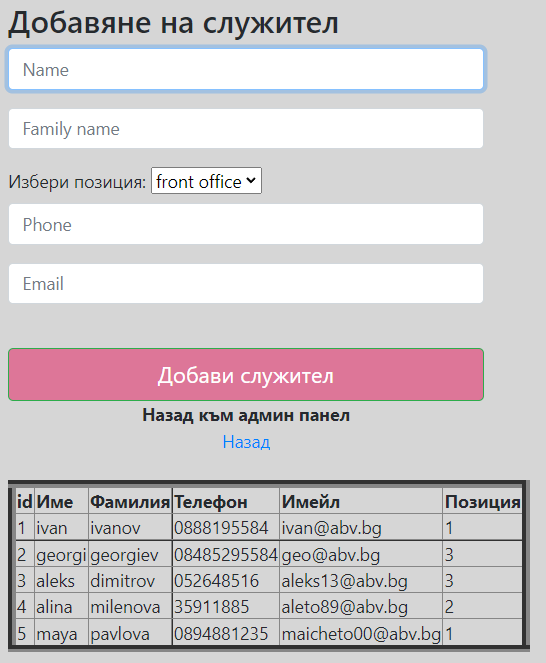
**39. Админски панел**



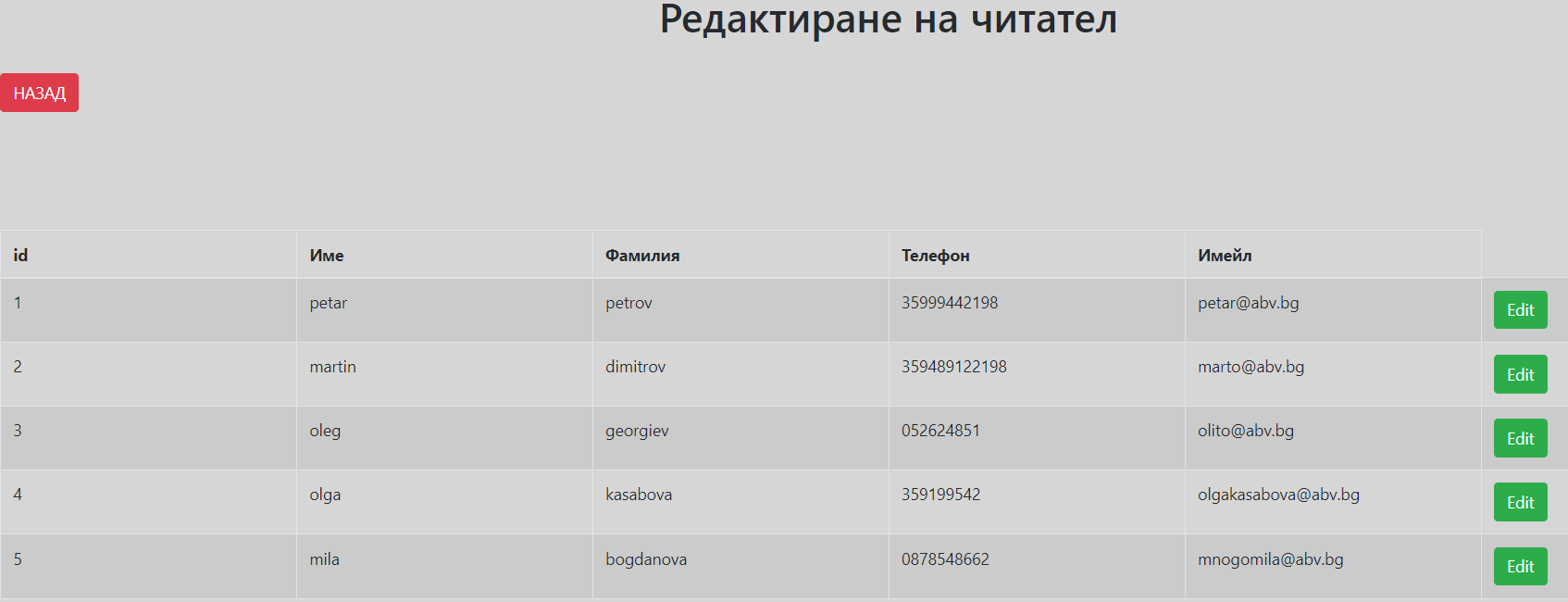
**40. Добавяне на читател от администратор**



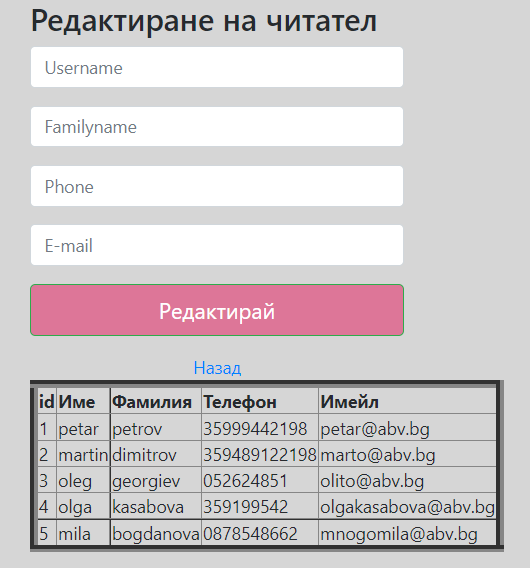
**41. Добавяне на служител от администратор**



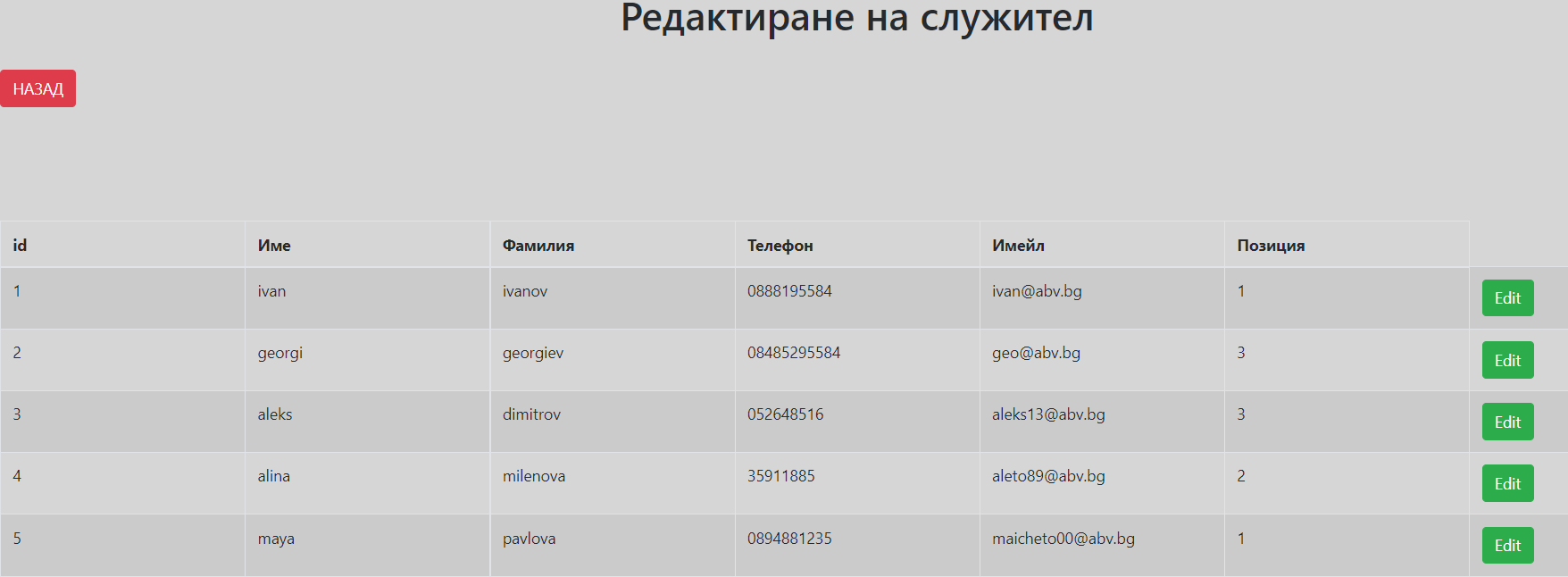
**42. Избор на читател за редактиране от администратор**



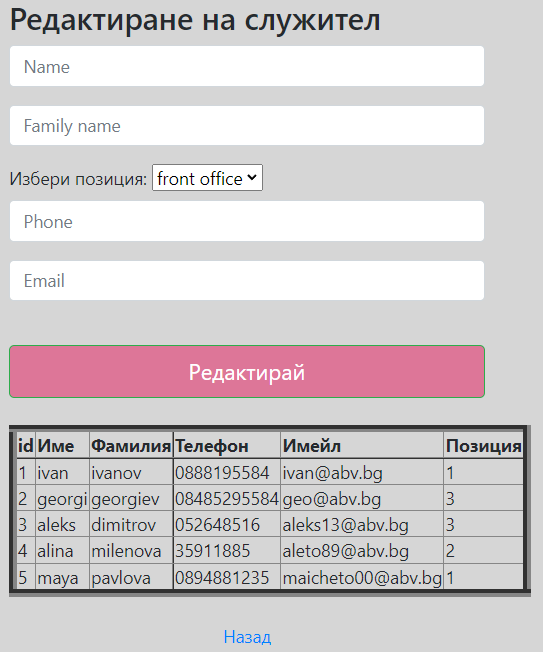
**43. Редактиране на данни на читател**



**44. Избор на служител за редактиране от администратор**



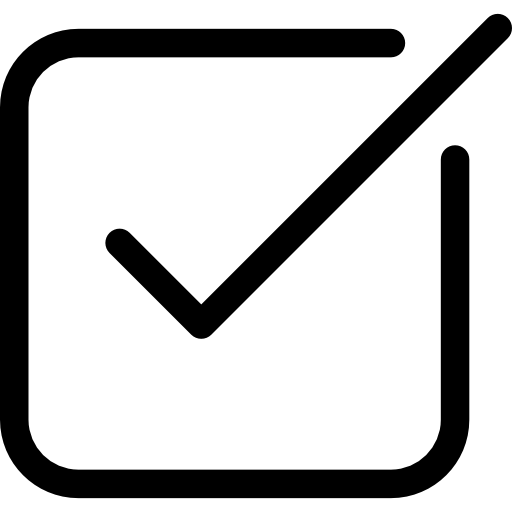
**45. Редактиране на данни на служител**



**V Перспективи за развитие**

Създадената система – библиотека, осигурява лесен достъп до функциите си за крайният потребител. Дизайнът е направен разбираем за използване. Уеб базираната система може да бъде развита още в следните направления:

* Създаване на мобилно приближение към уеб сайта с цел мобилност
* Подобряване сигурността на системата
* Добавяне на различни езици, за да имат достъп до него и чужди граждани



**VI Изходен код на системата**

**1. index.php – заглавна страница**

<!doctype html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<meta name="description" content="">

<meta name="author" content="">

<link rel="icon" href="/docs/4.0/assets/img/favicons/favicon.ico">

<title>Начало</title>

<link rel="canonical" href="https://getbootstrap.com/docs/4.0/examples/album/">

<link href="./bootstrap-4.0-dist/css/bootstrap.min.css" rel="stylesheet">

<link href="album.css" rel="stylesheet">

</head>

<body>

<header>

<div class="navbar navbar-dark bg-dark box-shadow">

<div class="container d-flex justify-content-between">

<a href="index.php" class="navbar-brand d-flex align-items-center">

<strong>Библиотека</strong>

</a>

<a href="./about.html" class="navbar-brand d-flex align-items-center">

<strong>За нас</strong>

</a>

</div>

</div>

</header>

<main role="main">

<section class="jumbotron text-center">

<div class="container">

<h1 class="jumbotron-heading">Библиотека - Явор Стоянов СИТ 2к 4Б група 20621620</h1>

<p class="lead text-muted">Мини проект WEB програмиране - библиотека</p>

<p>

<a href="./users/signin/readerUser.php" class="btn btn-primary my-2">Влез тук</a>

<a href="./users/registration/userRegistration.php" class="btn btn-secondary my-2">Регистрация</a>

</p>

</div>

</section>

<div class="album py-5 bg-light">

<div class="container">

<div class="row">

<div class="col-md-4">

<div class="card mb-4 box-shadow">

<div class="card-body">

<p class="card-text">Вход за читатели - petar@abv.bg petrov</p>

<div class="d-flex justify-content-between align-items-center">

<div class="btn-group">

<a href="./users/signin/readerUser.php" class="btn btn-secondary my-2">Влез</a>

</div>

</div>

</div>

</div>

</div>

<div class="col-md-4">

<div class="card mb-4 box-shadow">

<div class="card-body">

<p class="card-text">Вход за служители + справки - ivan ivanov</p>

<div class="d-flex justify-content-between align-items-center">

<div class="btn-group">

<a href="./users/signin/employeeUser.php" class="btn btn-secondary my-2">Влез</a>

</div>

</div>

</div>

</div>

</div>

<div class="col-md-4">

<div class="card mb-4 box-shadow">

<div class="card-body">

<p class="card-text">Вход за администратор - admin admin</p>

<div class="d-flex justify-content-between align-items-center">

<div class="btn-group">

<a href="./users/signin/admin.php" class="btn btn-secondary my-2">Влез</a>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</main>

</body>

</html>

<?php

include './db/db\_create.php';

include './db/config.php';

?>

**2. readerUser.php – влизане на потребител в системата**

<?php

session\_start();

?>

<html>

<head lang="en">

<meta charset="UTF-8">

<link type="text/css" rel="stylesheet" href="../../bootstrap-4.0-dist\css\bootstrap.css">

<title>Вход читател</title>

</head>

<style>

body{

background-image: url("img/background1.jpg");

background-repeat: no-repeat;

background-attachment: fixed;

background-size: 100% 100%;

background-color: lightgray;

}

.login-panel {

margin-top: 250px;

}

</style>

<body>

<div class="container">

<div class="col-md-4 col-md-offset-4">

<div class="login-panel">

<div class="panel-heading">

<h3 class="panel-title">Читател вход</h3>

</div>

<div class="panel-body">

<form role="form" method="post" action="#">

<fieldset>

<div class="form-group" >

<input class="form-control" placeholder="Email" name="red\_name" type="email" autofocus>

</div>

<div class="form-group">

<input class="form-control" placeholder="Password" name="red\_pass" type="password" value="">

</div>

<input class="btn btn-lg btn-success btn-block" type="submit" value="Влез" name="red\_login" style="background-color: palevioletred;" >

</fieldset>

</form>

<center><b>Назад към Начало</b> <br></b><a href="../../index.php">Начало</a> <br></b> </center>

</div>

</div>

</div>

</div>

</body>

<?php

include '../../db/config.php';

if(isset($\_POST['red\_login']))

{

$red\_name=$\_POST['red\_name'];

$red\_pass=$\_POST['red\_pass'];

$red\_query = "SELECT \* FROM reader WHERE reader\_email = '$red\_name' AND reader\_lName = '$red\_pass'";

$get\_id\_query = "SELECT id\_reader FROM reader WHERE reader\_email = '$red\_name' AND reader\_lName = '$red\_pass'";

$run\_query=mysqli\_query($dbConn,$red\_query);

$run\_get\_id = mysqli\_query($dbConn, $get\_id\_query);

$row = mysqli\_fetch\_row($run\_get\_id);

if($run\_query == TRUE){

if(mysqli\_num\_rows($run\_query)>0)

{

$\_SESSION['red\_name']=$red\_name;

$\_SESSION['red\_id'] = $row[0];

echo "<script>window.open('../readerpanel/readerpanel.php','\_self')</script>";

}

else {echo"<script>alert('Reader Details are incorrect!')</script>";}

} else {

die ("<br>Грешка: '. $run\_query->error");

}

}

**3. logout.php - Излизане на потребител / служител / админ от системата**

<?php

session\_start();

session\_destroy();

header("Location: ../../index.php");

?>

**4. userRegistration.php – регистрация на нов потребител**

<html>

<head lang="en">

<meta charset="UTF-8">

<link type="text/css" rel="stylesheet" href="../../bootstrap-4.0-dist\css\bootstrap.css">

<title>Регистрация</title>

</head>

<style>

body{

background-image: url("img/background1.jpg");

background-repeat: no-repeat;

background-attachment: fixed;

background-size: 100% 100%;

background-color: lightgray;

}

.login-panel {

margin-top: 250px;

}

</style>

<body>

<div class="container">

<div class="col-md-4 col-md-offset-4">

<div class="login-panel">

<div class="panel-heading">

<h3 class="panel-title">Регистрация</h3>

</div>

<div class="panel-body">

<form role="form" method="post" action="#">

<fieldset>

<div class="form-group">

<input class="form-control" placeholder="Username" name="name" type="text" autofocus required="">

</div>

<div class="form-group">

<input class="form-control" placeholder="Familyname" name="fname" type="text" value="" required="">

</div>

<div class="form-group">

<input class="form-control" placeholder="Phone" name="phone" type="text" value="" required="">

</div>

<div class="form-group">

<input class="form-control" placeholder="E-mail" name="email" type="email" autofocus required="">

</div>

<input class="btn btn-lg btn-success btn-block" type="submit" value="Регистрация" name="register" style="background-color: palevioletred;" >

</fieldset>

</form>

<center><b>Назад към Начало</b> <br></b><a href="../../index.php">Начало</a> <br></b> </center>

<center><b>Вече имаш акаунт?</b> <br></b><a href="../signin/readerUser.php">Влез тук</a></center>

</div>

</div>

</div>

</div>

</body>

<?php

include '../../db/config.php';

if(isset($\_POST['register']))

{

$user\_name = $\_POST['name'];

$user\_fname = $\_POST['fname'];

$user\_phone = $\_POST['phone'];

$user\_email = $\_POST['email'];

$check\_phone\_query="select \* from reader WHERE reader\_telNumber='$user\_phone'";

$run\_query1=mysqli\_query($dbConn,$check\_phone\_query);

if(mysqli\_num\_rows($run\_query1)>0)

{

echo "<script>alert('Phone is already exist in our database, Please try another one!')</script>";

exit();

}

$check\_email\_query="select \* from reader WHERE reader\_email='$user\_email'";

$run\_query2=mysqli\_query($dbConn,$check\_email\_query);

if(mysqli\_num\_rows($run\_query2)>0)

{

echo "<script>alert('Email is already exist in our database, Please try another one!')</script>";

exit();

}

$insertReader = "INSERT INTO libraryweb.reader(reader\_fName, reader\_lName, reader\_telNumber, reader\_email )

VALUES('$user\_name', '$user\_fname', '$user\_phone', '$user\_email') ";

if(mysqli\_query($dbConn,$insertReader))

{

echo"<script>window.open('../signin/readerUser.php','\_self')</script>";

}

}

**5. addBook.php – добавяне на книга от служител**

<?php

include '../../../db/config.php';

$sqlgen = "select \* from genre";

$all\_genre = mysqli\_query($dbConn, $sqlgen);

$sqlstat = "select \* from status";

$all\_status = mysqli\_query($dbConn, $sqlstat);

if(isset($\_POST['add']))

{

$title = mysqli\_real\_escape\_string($dbConn,$\_POST['title']);

$author = mysqli\_real\_escape\_string($dbConn,$\_POST['author']);

$year = mysqli\_real\_escape\_string($dbConn,$\_POST['year']);

$publisher = mysqli\_real\_escape\_string($dbConn,$\_POST['publisher']);

$genre = mysqli\_real\_escape\_string($dbConn,$\_POST['Genre']);

$status = mysqli\_real\_escape\_string($dbConn,$\_POST['Status']);

$takes = 0;

$check\_title\_query="select \* from book WHERE title='$title'";

$run\_query=mysqli\_query($dbConn,$check\_title\_query);

if(mysqli\_num\_rows($run\_query)>0)

{

echo "<script>alert('Book is already exist in our database, Please try another one!')</script>";

exit();

}

$insert\_book = "INSERT INTO book( title, author, yearOfPublishing, publisher, numberOfTakes, id\_genre, id\_status )

VALUES('$title', '$author', '$year', '$publisher', '$takes', '$genre', '$status')";

if ($dbConn->query($insert\_book) == TRUE)

{ echo '<script>alert("Book added successfully")</script>';}

else

{ echo '<script>alert('. $dbConn->error . ')</script>';}

}

?>

<html>

<head lang="en">

<meta charset="UTF-8">

<link type="text/css" rel="stylesheet" href="../../../bootstrap-4.0-dist\css\bootstrap.css">

<title>Добавяне на книга</title>

</head>

<style>

body{

background-image: url("img/background1.jpg");

background-repeat: no-repeat;

background-attachment: fixed;

background-size: 100% 100%;

background-color: lightgrey;

}

.login-panel {

margin-top: 25px;

}

</style>

<body>

<div class="container">

<div class="col-md-5 col-md-offset-3">

<div class="login-panel">

<div class="panel-heading">

<h3 class="panel-title">Добавяне на книга</h3>

</div>

<div class="panel-body">

<form role="form" method="post" action="#">

<fieldset>

<div class="form-group">

<input class="form-control" placeholder="Title" name="title" type="text" autofocus required="">

</div>

<div class="form-group">

<input class="form-control" placeholder="Author" name="author" type="text" autofocus required="">

</div>

<div class="form-group">

<input class="form-control" placeholder="Year of publishing" name="year" type="text" autofocus required="">

</div>

<div class="form-group">

<input class="form-control" placeholder="Publisher" name="publisher" type="text" autofocus required="">

</div>

<label>Избери жанр: </label>

<select name="Genre">

<?php

while ($gen = mysqli\_fetch\_array(

$all\_genre,MYSQLI\_ASSOC)):;

?>

<option value="<?php echo $gen["id\_genre"];

?>">

<?php echo $gen["genre"];

?>

</option>

<?php

endwhile;

?>

</select> <br><br>

<label>Избери статус: </label>

<select name="Status">

<?php

while ($stat = mysqli\_fetch\_array(

$all\_status,MYSQLI\_ASSOC)):;

?>

<option value="<?php echo $stat["id\_status"];

?>">

<?php echo $stat["status"];

?>

</option>

<?php

endwhile;

?>

</select> <br><br>

<input class="btn btn-lg btn-success btn-block" type="submit" value="Добави книга" name="add" style="background-color: palevioletred;" >

</fieldset>

</form>

<center><b>Назад към служител панел</b> <br></b><a href="./addpanel.php">Назад</a> <br></b> </center>

<table border="7">

<thead>

<tr>

<th>id</th>

<th>Заглавие</th>

<th>Автор</th>

<th>Година</th>

<th>Издател</th>

<th>Жанр</th>

<th>Брой заемания</th>

<th>Статус</th>

</tr>

</thead>

<?php

include '../../../db/config.php';

$view\_orders\_query="select \* from book";

$run=mysqli\_query($dbConn,$view\_orders\_query);

while($row=mysqli\_fetch\_array($run))

{

$book\_id=$row[0];

$book\_title=$row[1];

$book\_author=$row[2];

$book\_year=$row[3];

$book\_publisher=$row[4];

$book\_takes = $row[5];

$book\_genre = $row[6];

$book\_status = $row[7];

?>

<tr>

<td><?php echo $book\_id; ?></td>

<td><?php echo $book\_title; ?></td>

<td><?php echo $book\_author; ?></td>

<td><?php echo $book\_year; ?></td>

<td><?php echo $book\_publisher; ?></td>

<td><?php echo $book\_genre; ?></td>

<td><?php echo $book\_takes; ?></td>

<td><?php echo $book\_status; ?></td>

</tr>

<?php } ?>

</table>

</div>

</div>

</div>

</div>

</body>

</html>

**6. delBoo.php – избор на книга за изтриване от служител**

<html>

<head lang="en">

<meta charset="UTF-8">

<link type="text/css" rel="stylesheet" href="../../../../bootstrap-4.0-dist/css/bootstrap.css">

<title>Изтрий книга</title>

</head>

<style>

.login-panel {

margin-top: 50px;

}

.table {

margin-top: 100px;

}

body{

background-color: lightgrey;

background-image: url(../img/admin.jpg);

}

</style>

<body>

<div class="table-scrol">

<h1 align="center">Изтрий книга</h1><br>

<p><a align="center" href="../delpanel.php"><button class="btn btn-danger">НАЗАД</button></a></p>

<div class="table-responsive">

<table class="table table-bordered table-hover table-striped" style="table-layout: fixed">

<thead>

<tr>

<th>id</th>

<th>Заглавие</th>

<th>Автор</th>

<th>Година</th>

<th>Издател</th>

<th>Жанр</th>

<th>Брой заемания</th>

<th>Статус</th>

</tr>

</thead>

<?php

include '../../../../db/config.php';

$view\_book\_query="select \* from book";

$run=mysqli\_query($dbConn,$view\_book\_query);

while($row=mysqli\_fetch\_array($run))

{

$book\_id=$row[0];

$book\_title=$row[1];

$book\_author=$row[2];

$book\_year=$row[3];

$book\_publisher=$row[4];

$book\_takes = $row[5];

$book\_genre = $row[6];

$book\_status = $row[7];

?>

<tr>

<td><?php echo $book\_id; ?></td>

<td><?php echo $book\_title; ?></td>

<td><?php echo $book\_author; ?></td>

<td><?php echo $book\_year; ?></td>

<td><?php echo $book\_publisher; ?></td>

<td><?php echo $book\_genre; ?></td>

<td><?php echo $book\_takes; ?></td>

<td><?php echo $book\_status; ?></td>

<td><a href="delB.php?del=<?php echo $book\_id ?>"><button class="btn btn-danger">Delete</button></a></td>

</tr>

<?php } ?>

</table>

</div>

</div>

</body>

</html>

**7. delB.php – изтриване на избрана книга от служител**

<?php

include '../../../../db/config.php';

$delete\_id=$\_GET['del'];

$delete\_query="delete from book WHERE id\_book='$delete\_id'";

$run=mysqli\_query($dbConn,$delete\_query);

if($run)

{

echo "<script>window.open('delBoo.php?deleted=book has been deleted','\_self')</script>";

}

?>

**8. spravka1.php - Справка за не върнати книги подредени по дата на връщане**

<html>

<head lang="en">

<meta charset="UTF-8">

<link type="text/css" rel="stylesheet" href="../../../bootstrap-4.0-dist/css/bootstrap.css">

<title>Невърнати книги по дата</title>

</head>

<style>

.login-panel {

margin-top: 50px;

}

.table {

margin-top: 100px;

}

body{

background-color: lightgrey;

background-image: url(../img/admin.jpg);

}

</style>

<body>

<div class="table-scrol">

<h1 align="center">Невърнати книги подредени по дата на връщане</h1><br>

<p><a align="center" href="../employeepanel.php"><button class="btn btn-danger">НАЗАД</button></a></p>

<div class="table-responsive">

<table class="table table-bordered table-hover table-striped" style="table-layout: fixed">

<thead>

<tr>

<th>Заглавие</th>

<th>Срок за връщане</th>

</tr>

</thead>

<?php

include '../../../db/config.php';

$view\_book\_query="SELECT b.title, z.srokVryshtane FROM zaemane z INNER JOIN book b ON z.book\_id = b.id\_book WHERE z.knigaVryshtane = 0 ORDER BY z.srokVryshtane ASC";

$run=mysqli\_query($dbConn,$view\_book\_query);

while($row=mysqli\_fetch\_array($run))

{

$book\_id=$row[0];

$book\_title=$row[1];

?>

<tr>

<td><?php echo $book\_id; ?></td>

<td><?php echo $book\_title; ?></td>

</tr>

<?php } ?>

</table>

</div>

</div>

</body>

</html>

**9. spravka2.php – свободни книги подредени по брой заемания**

<html>

<head lang="en">

<meta charset="UTF-8">

<link type="text/css" rel="stylesheet" href="../../../bootstrap-4.0-dist/css/bootstrap.css">

<title>Свободни книги по брой заемания</title>

</head>

<style>

.login-panel {

margin-top: 50px;

}

.table {

margin-top: 100px;

}

body{

background-color: lightgrey;

background-image: url(../img/admin.jpg);

}

</style>

<body>

<div class="table-scrol">

<h1 align="center">Свободни книги подредени по брой заемания</h1><br>

<p><a align="center" href="../employeepanel.php"><button class="btn btn-danger">НАЗАД</button></a></p>

<div class="table-responsive">

<table class="table table-bordered table-hover table-striped" style="table-layout: fixed">

<thead>

<tr>

<th>Заглавие</th>

<th>Брой заемания</th>

</tr>

</thead>

<?php

include '../../../db/config.php';

$view\_book\_query="SELECT title, numberOfTakes FROM book WHERE id\_status = 1 ORDER BY numberOfTakes DESC";

$run=mysqli\_query($dbConn,$view\_book\_query);

while($row=mysqli\_fetch\_array($run))

{

$book\_id=$row[0];

$book\_title=$row[1];

?>

<tr>

<td><?php echo $book\_id; ?></td>

<td><?php echo $book\_title; ?></td>

</tr>

<?php } ?>

</table>

</div>

</div>

</body>

</html>

**10. spravka3.php – най- често заемани книги**

<html>

<head lang="en">

<meta charset="UTF-8">

<link type="text/css" rel="stylesheet" href="../../../bootstrap-4.0-dist/css/bootstrap.css">

<title>Най-често заемани книги</title>

</head>

<style>

.login-panel {

margin-top: 50px;

}

.table {

margin-top: 100px;

}

body{

background-color: lightgrey;

background-image: url(../img/admin.jpg);

}

</style>

<body>

<div class="table-scrol">

<h1 align="center">Най-често заемани книги</h1><br>

<p><a align="center" href="../employeepanel.php"><button class="btn btn-danger">НАЗАД</button></a></p>

<div class="table-responsive">

<table class="table table-bordered table-hover table-striped" style="table-layout: fixed">

<thead>

<tr>

<th>Заглавие</th>

<th>Честота на заемане</th>

</tr>

</thead>

<?php

include '../../../db/config.php';

$view\_book\_query="SELECT b.title, COUNT(z.book\_id) AS MOST\_FREQUENT

FROM zaemane z INNER JOIN book b ON z.book\_id = b.id\_book

GROUP BY book\_id

ORDER BY COUNT(book\_id) DESC";

$run=mysqli\_query($dbConn,$view\_book\_query);

while($row=mysqli\_fetch\_array($run))

{

$book\_id=$row[0];

$book\_title=$row[1];

?>

<tr>

<td><?php echo $book\_id; ?></td>

<td><?php echo $book\_title; ?></td>

</tr>

<?php } ?>

</table>

</div>

</div>

</body>

</html>

**11. spravka4.php – топ 5 читатели по брой заети книги**

<html>

<head lang="en">

<meta charset="UTF-8">

<link type="text/css" rel="stylesheet" href="../../../bootstrap-4.0-dist/css/bootstrap.css">

<title>Toп 5 читатели по брой заети книги</title>

</head>

<style>

.login-panel {

margin-top: 50px;

}

.table {

margin-top: 100px;

}

body{

background-color: lightgrey;

background-image: url(../img/admin.jpg);

}

</style>

<body>

<div class="table-scrol">

<h1 align="center">Toп 5 читатели по брой заети книги</h1><br>

<p><a align="center" href="../employeepanel.php"><button class="btn btn-danger">НАЗАД</button></a></p>

<div class="table-responsive">

<table class="table table-bordered table-hover table-striped" style="table-layout: fixed">

<thead>

<tr>

<th>Пълно име</th>

<th>Брой заети книги</th>

</tr>

</thead>

<?php

include '../../../db/config.php';

$view\_book\_query="SELECT CONCAT(r.reader\_fName,' ', r.reader\_lName) AS fullname, COUNT(z.id\_reader) AS MOST\_FREQUENT

FROM zaemane z INNER JOIN reader r ON z.id\_reader = r.id\_reader

GROUP BY z.id\_reader

ORDER BY COUNT(z.id\_reader) DESC

LIMIT 5";

$run=mysqli\_query($dbConn,$view\_book\_query);

while($row=mysqli\_fetch\_array($run))

{

$book\_id=$row[0];

$book\_title=$row[1];

?>

<tr>

<td><?php echo $book\_id; ?></td>

<td><?php echo $book\_title; ?></td>

</tr>

<?php } ?>

</table>

</div>

</div>

</body>

</html>

**12. config.php – връзка с базата данни**

<?php

$host = 'localhost';

$dbUser = 'root';

$dbPass = '';

$dbName = 'libraryweb';

$tableBook = 'book';

$tableEmployee = 'employee';

$tableGenre = 'genre';

$tablePositions = 'positions';

$tableReader = 'reader';

$tableStatus = 'status';

$tableZaemane = 'zaemane';

$dbConn=mysqli\_connect($host, $dbUser, $dbPass);

if (!$dbConn=mysqli\_connect($host, $dbUser, $dbPass)) {

die('Няма връзка със сървър');

} else {

// echo "<br> има връзка със сървър";

}

if (!mysqli\_select\_db($dbConn,$dbName)) {

die('Грешка при селектиране');

} else {

// echo "<br> селектиране на бд бачка";

}

mysqli\_query($dbConn,"SET NAMES 'UTF8'");

?>

**13. db\_create.php - скрипт за създаване на базата данни, таблиците в нея и тестови данни**

<?php

$conn = new mysqli('localhost', 'root', '');

if ($conn->connect\_error){

die('<br>Грешка: '. $conn->connect\_error);

}

$sql = "CREATE DATABASE libraryweb";

if ($conn->query($sql) == TRUE){

// echo '<br>БД libraryweb e създадена<br>';

}

else {

// echo '<br>Грешка: '. $conn->error;

}

//book table

$sqlBook = "CREATE TABLE libraryweb.book (

id\_book int(5) NOT NULL AUTO\_INCREMENT,

title varchar(80) NOT NULL,

author varchar(50) NOT NULL,

yearOfPublishing int(5) NOT NULL,

publisher varchar(20) NOT NULL,

numberOfTakes int(5) NOT NULL,

PRIMARY KEY (`id\_book`));";

//id\_genre int(5) NOT NULL, id\_status int(10) NOT NULL,

if ($conn->query($sqlBook) == TRUE)

{// echo '<br>Tаблица book е създадена';

}

else

{// echo '<br>1Грешка: '. $conn->error;

}

//employee table

$sqlEmployee = "CREATE TABLE libraryweb.employee (

id\_employee int(5) NOT NULL AUTO\_INCREMENT,

employee\_fName varchar(20) NOT NULL,

employee\_lName varchar(20) NOT NULL,

employee\_telNumber varchar(20) NOT NULL,

employee\_email varchar(20) NOT NULL,

PRIMARY KEY (`id\_employee`));";

if ($conn->query($sqlEmployee) == TRUE)

{ //echo '<br>Tаблица employee е създадена';

}

else

{// echo '<br>2Грешка: '. $conn->error;

}

//genre table

$sqlGenre = "CREATE TABLE libraryweb.genre (

id\_genre int(5) NOT NULL AUTO\_INCREMENT,

genre varchar(20) NOT NULL,

PRIMARY KEY (`id\_genre`));";

if ($conn->query($sqlGenre) == TRUE)

{ //echo '<br>Tаблица genre е създадена';

}

else

{ //echo '<br>3Грешка: '. $conn->error;

}

//positions table

$sqlPositions = "CREATE TABLE libraryweb.positions (

id\_positions int(5) NOT NULL AUTO\_INCREMENT,

positions varchar(20) NOT NULL,

PRIMARY KEY (`id\_positions`));";

if ($conn->query($sqlPositions) == TRUE)

{ //echo '<br>Tаблица positions е създадена';

}

else

{ //echo '<br>4Грешка: '. $conn->error;

}

//reader table

$sqlReader = "CREATE TABLE libraryweb.reader (

id\_reader int(5) NOT NULL AUTO\_INCREMENT,

reader\_fName varchar(20) NOT NULL,

reader\_lName varchar(20) NOT NULL,

reader\_telNumber varchar(20) NOT NULL,

reader\_email varchar(20) NOT NULL,

PRIMARY KEY (`id\_reader`));";

if ($conn->query($sqlReader) == TRUE)

{ //echo '<br>Tаблица reader е създадена';

}

else

{ //echo '<br>5Грешка: '. $conn->error;

}

//status table

$sqlStatus = "CREATE TABLE libraryweb.status (

id\_status int(10) NOT NULL AUTO\_INCREMENT,

status varchar(20) NOT NULL,

PRIMARY KEY (`id\_status`));";

if ($conn->query($sqlStatus) == TRUE)

{ //echo '<br>Tаблица status е създадена';

}

else

{ //

//echo '<br>6Грешка: '. $conn->error;

}

//zaemane table

$sqlZaemane = "CREATE TABLE libraryweb.zaemane (

id\_zaemane int(11) NOT NULL AUTO\_INCREMENT,

dataZaemane datetime NOT NULL DEFAULT current\_timestamp(),

srokVryshtane timestamp NOT NULL DEFAULT current\_timestamp() ON UPDATE current\_timestamp(),

knigaVryshtane datetime NOT NULL ,

PRIMARY KEY (`id\_zaemane`));";

//dataZaemane date NOT NULL,

//srokVryshtane date NOT NULL,

if ($conn->query($sqlZaemane) == TRUE)

{ //echo '<br>Tаблица zaemane е създадена';

}

else

{ //echo '<br>7Грешка: '. $conn->error;

}

//book FK

$sqlBookFK = "ALTER TABLE libraryweb.book

ADD id\_genre int(5) NOT NULL,

ADD id\_status int(5) NOT NULL,

ADD CONSTRAINT book\_genre\_f\_key FOREIGN KEY (`id\_genre`) REFERENCES libraryweb.genre (`id\_genre`) ON DELETE

CASCADE ON UPDATE CASCADE,

ADD CONSTRAINT book\_status\_f\_key FOREIGN KEY (`id\_status`) REFERENCES libraryweb.status (`id\_status`) ON DELETE

CASCADE ON UPDATE CASCADE;";

if ($conn->query($sqlBookFK) == TRUE)

{ //echo '<br>FK за book е създаден';

}

else

{ //echo '<br>8Грешка: '. $conn->error;

}

//employee FK

$sqlEmployeeFK = "ALTER TABLE libraryweb.employee

ADD id\_positions int(5) NOT NULL,

ADD CONSTRAINT employee\_positions\_f\_key FOREIGN KEY (`id\_positions`) REFERENCES libraryweb.positions (`id\_positions`) ON DELETE

CASCADE ON UPDATE CASCADE;";

if ($conn->query($sqlEmployeeFK) == TRUE)

{ //echo '<br>FK за employee е създаден';

}

else

{ //echo '<br>###Грешка: '. $conn->error;

}

//zaemane FK

$sqlZaemaneFK = "ALTER TABLE libraryweb.zaemane

ADD book\_id int(5) NOT NULL,

ADD id\_reader int(5) NOT NULL,

ADD CONSTRAINT zaemane\_book\_f\_key FOREIGN KEY (`book\_id`) REFERENCES libraryweb.book (`id\_book`) ON DELETE

CASCADE ON UPDATE CASCADE,

ADD CONSTRAINT zaemane\_reader\_f\_key FOREIGN KEY (`id\_reader`) REFERENCES libraryweb.reader (`id\_reader`) ON DELETE

CASCADE ON UPDATE CASCADE;";

//ADD id\_employee int(5) NOT NULL,

// ADD CONSTRAINT zaemane\_employee\_f\_key FOREIGN KEY (`id\_employee`) REFERENCES libraryweb.employee (`id\_employee`) ON DELETE

// CASCADE ON UPDATE CASCADE,

if ($conn->query($sqlZaemaneFK) == TRUE)

{// echo '<br>FK за zaemane е създаден';

}

else

{ //echo '<br>10Грешка: '. $conn->error;

}

//setvame statusa

$sqlStatusSet1 = "INSERT INTO libraryweb.status(id\_status, status)

VALUES(1, 'svobodna');";

$sqlStatusSet2 = "INSERT INTO libraryweb.status(id\_status, status)

VALUES(2, 'zaeta');";

if ($conn->query($sqlStatusSet1) == TRUE)

{ //echo '<br>статуса е сетнат 1 svobodna';

}

else

{ //echo '<br>11Грешка: '. $conn->error;

}

if ($conn->query($sqlStatusSet2) == TRUE)

{ //echo '<br>статуса е сетнат 2 zaeta';

}

else

{ //echo '<br>12Грешка: '. $conn->error;

}

//admin table

$sqlAdmin = "CREATE TABLE libraryweb.admin (

id\_admin int(11) NOT NULL AUTO\_INCREMENT,

admin\_name varchar(20),

admin\_pass varchar(20),

PRIMARY KEY (`id\_admin`));";

if ($conn->query($sqlAdmin) == TRUE)

{ //echo '<br>Tаблица admin е създадена';

}

else

{ //echo '<br>13Грешка: '. $conn->error;

}

$sqlPositionsAdmin = "INSERT INTO libraryweb.positions(id\_positions, positions)

VALUES(1, 'front office')";

$sqlPositionsSecretary = "INSERT INTO libraryweb.positions(id\_positions, positions)

VALUES(2, 'secretary')";

$sqlPositionsDirector = "INSERT INTO libraryweb.positions(id\_positions, positions)

VALUES(3, 'director')";

//setvame 1 "admin"

$sqlEmployeeAdmin = "INSERT INTO libraryweb.admin(id\_admin, admin\_name, admin\_pass)

VALUES(1, 'admin', 'admin')";

$sqlscifi = "INSERT INTO libraryweb.genre(id\_genre, genre) VALUES(1 , 'scifi')";

$sqlhorror = "INSERT INTO libraryweb.genre(id\_genre, genre) VALUES(2 , 'horror')";

$sqlfantasy = "INSERT INTO libraryweb.genre(id\_genre, genre) VALUES(3 , 'fantasy')";

if ($conn->query($sqlPositionsAdmin) == TRUE)

{ //echo '<br>position fron office е сетната';

}

else

{ //echo '<br>14Грешка: '. $conn->error;

}

if ($conn->query($sqlEmployeeAdmin) == TRUE)

{ //echo '<br>admin е сетнат';

}

else

{ //echo '<br>15Грешка: '. $conn->error;

}

if ($conn->query($sqlPositionsSecretary) == TRUE)

{ //echo '<br>secretary е сетнат';

}

else

{ //echo '<br>16Грешка: '. $conn->error;

}

if ($conn->query($sqlscifi) == TRUE)

{ //echo '<br>scifi е сетнат';

}

else

{

//echo '<br>17Грешка: '. $conn->error;

}

if ($conn->query($sqlhorror) == TRUE)

{ //echo '<br>horror е сетнат';

}

else

{ //echo '<br>18Грешка: '. $conn->error;

}

if ($conn->query($sqlPositionsDirector) == TRUE)

{ //echo '<br>deirector е сетнат';

}

else

{ //echo '<br>19Грешка: '. $conn->error;

}

if ($conn->query($sqlfantasy) == TRUE)

{ //echo '<br>fantasy е сетнат';

}

else

{ //echo '<br>20Грешка: '. $conn->error;

}

$sqlTestEmpl1 = "INSERT INTO libraryweb.employee(id\_employee, employee\_fName, employee\_lName, employee\_telNumber, employee\_email, id\_positions)

VALUES(1, 'ivan', 'ivanov', '0888195584', 'ivan@abv.bg', 1)";

$sqlTestReader1 = "INSERT INTO libraryweb.reader( id\_reader, reader\_fName, reader\_lName, reader\_telNumber, reader\_email )

VALUES(1, 'petar', 'petrov', '35999442198', 'petar@abv.bg') ";

if ($conn->query($sqlTestEmpl1) == TRUE)

{ //echo '<br>ivan empl е сетнат';

}

else

{ //echo '<br>21Грешка: '. $conn->error;

}

if ($conn->query($sqlTestReader1) == TRUE)

{ //echo '<br>petar reader е сетнат';

}

else

{ //echo '<br>22Грешка: '. $conn->error;

}

$sqlTestEmpl2 = "INSERT INTO libraryweb.employee(id\_employee, employee\_fName, employee\_lName, employee\_telNumber, employee\_email, id\_positions)

VALUES(2, 'georgi', 'georgiev', '08485295584', 'geo@abv.bg', 3)";

$sqlTestReader2 = "INSERT INTO libraryweb.reader( id\_reader, reader\_fName, reader\_lName, reader\_telNumber, reader\_email )

VALUES(2, 'martin', 'dimitrov', '359489122198', 'marto@abv.bg') ";

if ($conn->query($sqlTestEmpl2) == TRUE)

{ //echo '<br>georgi empl е сетнат';

}

else

{// echo '<br>23Грешка: '. $conn->error;

}

if ($conn->query($sqlTestReader2) == TRUE)

{ //echo '<br>martin reader е сетнат';

}

else

{ //echo '<br>24Грешка: '. $conn->error;

}

$sqlTestEmpl3 = "INSERT INTO libraryweb.employee(id\_employee, employee\_fName, employee\_lName, employee\_telNumber, employee\_email, id\_positions)

VALUES(3, 'aleks', 'dimitrov', '052648516', 'aleks13@abv.bg', 3)";

$sqlTestReader3 = "INSERT INTO libraryweb.reader( id\_reader, reader\_fName, reader\_lName, reader\_telNumber, reader\_email )

VALUES(3, 'oleg', 'georgiev', '052624851', 'olito@abv.bg') ";

if ($conn->query($sqlTestEmpl3) == TRUE)

{// echo '<br>aleks empl е сетнат';

}

else

{// echo '<br>25Грешка: '. $conn->error;

}

if ($conn->query($sqlTestReader3) == TRUE)

{// echo '<br>oleg reader е сетнат';

}

else

{ //echo '<br>26Грешка: '. $conn->error;

}

$sqlTestEmpl4 = "INSERT INTO libraryweb.employee(id\_employee, employee\_fName, employee\_lName, employee\_telNumber, employee\_email, id\_positions)

VALUES(4, 'alina', 'milenova', '35911885', 'aleto89@abv.bg', 2)";

$sqlTestReader4 = "INSERT INTO libraryweb.reader( id\_reader, reader\_fName, reader\_lName, reader\_telNumber, reader\_email )

VALUES(4, 'olga', 'kasabova', '359199542', 'olgakasabova@abv.bg') ";

if ($conn->query($sqlTestEmpl4) == TRUE)

{ //echo '<br>alina empl е сетнат';

}

else

{ //echo '<br>27Грешка: '. $conn->error;

}

if ($conn->query($sqlTestReader4) == TRUE)

{ //echo '<br>olga reader е сетнат';

}

else

{ //echo '<br>28Грешка: '. $conn->error;

}

$sqlTestEmpl5 = "INSERT INTO libraryweb.employee(id\_employee, employee\_fName, employee\_lName, employee\_telNumber, employee\_email, id\_positions)

VALUES(5, 'maya', 'pavlova', '0894881235', 'maicheto00@abv.bg', 1)";

$sqlTestReader5 = "INSERT INTO libraryweb.reader( id\_reader, reader\_fName, reader\_lName, reader\_telNumber, reader\_email )

VALUES(5, 'mila', 'bogdanova', '0878548662', 'mnogomila@abv.bg') ";

if ($conn->query($sqlTestEmpl5) == TRUE)

{ //echo '<br>maya empl е сетнат';

}

else

{// echo '<br>29Грешка: '. $conn->error;

}

if ($conn->query($sqlTestReader5) == TRUE)

{ //echo '<br>mila reader е сетнат';

}

else

{ //echo '<br>30Грешка: '. $conn->error;

}

$sqlbook1 = "INSERT INTO libraryweb.book(id\_book, title, author, yearOfPublishing, publisher, numberOfTakes, id\_genre, id\_status)

VALUES(1, 'Ad Astra', 'Jack Campbell', 2013, 'Amazon',0, 1,1)";

$sqlbook2 = "INSERT INTO libraryweb.book(id\_book, title, author, yearOfPublishing, publisher, numberOfTakes, id\_genre, id\_status)

VALUES(2, 'Project Hail Mary', 'Andy Weir', 2021, 'Weir',0, 1,1)";

$sqlbook3 = "INSERT INTO libraryweb.book(id\_book, title, author, yearOfPublishing, publisher, numberOfTakes, id\_genre, id\_status)

VALUES(3, 'Harry Potter: Chamber of Secrets', 'J. K. Rowling', 1998, 'J. K. Rowling',0, 3,1)";

$sqlbook4 = "INSERT INTO libraryweb.book(id\_book, title, author, yearOfPublishing, publisher, numberOfTakes, id\_genre, id\_status)

VALUES(4, 'Harry Potter: Prisoner of Azkaban', 'J. K. Rowling', 1999, 'J. K. Rowling',0, 3,1)";

$sqlbook5 = "INSERT INTO libraryweb.book(id\_book, title, author, yearOfPublishing, publisher, numberOfTakes, id\_genre, id\_status)

VALUES(5, 'Harry Potter: Goblet of Fire', 'J. K. Rowling', 2000, 'J. K. Rowling',0, 3,1)";

$sqlbook6 = "INSERT INTO libraryweb.book(id\_book, title, author, yearOfPublishing, publisher, numberOfTakes, id\_genre, id\_status)

VALUES(6, 'Harry Potter: Order of the Phoenix', 'J. K. Rowling', 2003, 'J. K. Rowling',0, 3,1)";

$sqlbook7 = "INSERT INTO libraryweb.book(id\_book, title, author, yearOfPublishing, publisher, numberOfTakes, id\_genre, id\_status)

VALUES(7, 'Dracula', 'Bram Stoker', 1897, 'Stoker',0, 2,1)";

$sqlbook8 = "INSERT INTO libraryweb.book(id\_book, title, author, yearOfPublishing, publisher, numberOfTakes, id\_genre, id\_status)

VALUES(8, 'Frankenstein', 'Mary Shelley', 1818, 'Shelley',0, 2,1)";

$sqlbook9 = "INSERT INTO libraryweb.book(id\_book, title, author, yearOfPublishing, publisher, numberOfTakes, id\_genre, id\_status)

VALUES(9, 'Dune', 'Frank Herbert', 1965, 'Hernert',0, 1,1)";

if ($conn->query($sqlbook1) == TRUE)

{ //echo '<br>position fron office е сетната';

}

else

{ //echo '<br>14Грешка: '. $conn->error;

}

if ($conn->query($sqlbook2) == TRUE)

{ //echo '<br>position fron office е сетната';

}

else

{ //echo '<br>14Грешка: '. $conn->error;

}

if ($conn->query($sqlbook3) == TRUE)

{ //echo '<br>position fron office е сетната';

}

else

{ //echo '<br>14Грешка: '. $conn->error;

}

if ($conn->query($sqlbook4) == TRUE)

{ //echo '<br>position fron office е сетната';

}

else

{ //echo '<br>14Грешка: '. $conn->error;

}

if ($conn->query($sqlbook5) == TRUE)

{ //echo '<br>position fron office е сетната';

}

else

{ //echo '<br>14Грешка: '. $conn->error;

}

if ($conn->query($sqlbook6) == TRUE)

{ //echo '<br>position fron office е сетната';

}

else

{ //echo '<br>14Грешка: '. $conn->error;

}

if ($conn->query($sqlbook7) == TRUE)

{ //echo '<br>position fron office е сетната';

}

else

{ //echo '<br>14Грешка: '. $conn->error;

}

if ($conn->query($sqlbook8) == TRUE)

{ //echo '<br>position fron office е сетната';

}

else

{ //echo '<br>14Грешка: '. $conn->error;

}

if ($conn->query($sqlbook9) == TRUE)

{ //echo '<br>position fron office е сетната';

}

else

{ //echo '<br>14Грешка: '. $conn->error;

}

$conn->close();

**VII Източници**

* [**https://bg.wikipedia.org/**](https://bg.wikipedia.org/)
* [**https://www.w3schools.com/**](https://www.w3schools.com/)
* [**https://getbootstrap.com/docs/5.0/getting-started/introduction/**](https://getbootstrap.com/docs/5.0/getting-started/introduction/)
* **Материали по дисциплина WEB програмиране, предоставени от доц. д-р инж Виолета Божикова и**

**гл. ас. д-р. инж. Диян Динев**