РЕФЕРАТ

Пояснювальна записка до курсового проекту на тему «Розробка інтернет-магазину книг» складається з переліку умовних скорочень, вступу, трьох розділів, висновків, списку використаної літератури та додатків.

Текстова частина викладена на 38 сторінках друкованого тексту.

Пояснювальна записка має 50 сторінок додатків. Список використаних джерел містить 10 найменувань і займає 1 сторінку. В роботі наведено 26 рисунків. Загальний обсяг роботи – 88 сторінок.

У першому розділі було обґрунтовано створення веб-додатку та загальні теоретичні питання, що стосуються заданої теми, для того, щоб встановити всі необхідні вимоги для майбутнього веб-сайту.

У другому розділі проведено проектування і розробка програмного продукту.

У третьому розділі проведено тестування веб-додатку.

Висновок містить в собі результати виконаної роботи створення веб-сайту.

У додатках представлений лістинг розробленого веб-сайту.

Ключові слова: ВЕБ-ДОДАТОК, ВЕБ-САЙТ, ІНТЕРНЕТ-МАГАЗИН, PHP, MVC, CMS

ЗМІСТ

[Перелік умовних скорочень 6](#_Toc516686107)

[ВСТУП 7](#_Toc516686108)

[РОЗДІЛ 1. АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ 8](#_Toc516686109)

[1.1 Аналіз задачі, засобів та методів її вирішення 8](#_Toc516686110)

[1.2 Аналіз існуючих веб-сайтів за тематикою курсового проекту 9](#_Toc516686111)

[1.3 Технічне завдання до курсовий проект 12](#_Toc516686112)

[Висновки до першого розділу 13](#_Toc516686113)

[РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РОЗРОБКА ВЕБ-САЙТУ 14](#_Toc516686114)

[2.1 Проектування загального алгоритму роботи веб-додатку 14](#_Toc516686115)

[2.2 Розробка функціональних алгоритмів роботи веб-сайту 16](#_Toc516686116)

[2.3 Розробка веб-додатку 18](#_Toc516686117)

[Висновки до другого розділу 25](#_Toc516686118)

[РОЗДІЛ 3. ОПИС РОБОТИ З ВЕБ-САЙТОМ ТА ЙОГО ТЕСТУВАННЯ 26](#_Toc516686119)

[3.1 Опис роботи з веб-сайтом 26](#_Toc516686120)

[3.2 Тестування веб-додатку 34](#_Toc516686121)

[Висновки до третього розділу 35](#_Toc516686122)

[ВИСНОВКИ 36](#_Toc516686123)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 37](#_Toc516686124)

# Перелік умовних скорочень

ВД – веб-додаток

ВС – веб-сайт

ПЗ – програмне забезпечення

ІМ – інтернет-магазин

OS – Open Server

CMS – Content Management System

JS - JavaScript

БД – база даних

# ВСТУП

В курсовому проекті розписано процес створення програмного забезпечення, яке включає в себе розробку багатосторінкового веб-додатку з використанням паттерну MVC та власною системою адміністрування.

Веб-додатки типу «інтернет-магазин» дають можливість здійснювати купівлю товарів з найрізноманітніших сфер людської діяльності через Інтернет. В даній роботі буде описана розробка інтернет-магазину з назвою «Букініст» для продажу та купівлі літератури.

Об’єктом дослідження роботи є створення CMS з можливістю додавання та редагування даних, а також створення різноманітних модулів сайту.

Предметом дослідження виступає веб-сайт, який графічно зображає створені модулі та систему управління, надає можливість реєстрації користувачів та здійснення покупок через сайт.

Мета дослідження – створення веб-додатку з коректно відображеними і функціонуючими сторінками, виконаними у зрозумілому для користувача інтерфейсі.

# РОЗДІЛ 1. АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ

## **Аналіз задачі, засобів та методів її вирішення**

Інтернет-магазини – розповсюджений тип веб-додатків, до яких звикли всі користувачі мережі інтернет. ІМ дозволяють купляти будь-які представлені в асортименті товари не виходячи з дому, за що і отримали свою популярність.

Найчастіше ІМ створюються на основі справжніх маркетів для збільшення їх популярності та переходу до більш сучасної варіації купівлі-продажу товарів.

Даний тип веб-додатку обов’язково містить головну сторінку та сторінку з каталогом товарів. Інші сторінки, наприклад «Контакти», «Відгуки» і т.д. додаються за бажанням, або розміщуються на основних сторінках. Для зручного переходу між сторінками роблять «шапку сайту», на якій розміщюються кнопки з назвами розділів сайту.

Існує доволі розповсюджена практика, коли користувачі отримують можливість купляти товари після реєстрації на сайті. Це робиться для зручного збору даних про користувачів веб-додатку та їх активність.

Товари додаються адміністраторами та контент-менеджерами, які мають особливі привілегії у користуванні веб-сайтом.

В роботі буде створюватись багатосторінковий веб-сайт з реєстрацією користувачів, каталогом та системою адміністрування. Буде створена панель для зручного переходу між сторінками. Це забезпечить розуміння користувачем інтерфейсу та можливостей веб-додатку.

Для перерахованих вище можливостей правильно буде використати паттерн MVC. Це особливий шаблон розподілення даних у вигляді «Model-view-controller», що передбачає створення трьох файлів, що виконують різні функції. Вони взаємодіють між собою та забезпечують коректну роботу додатків без плутанини розробника у власному коді. Паттерн забезпечує гнучкість та легкість змін у дизайні та функціонуванні певних частин програмних продуктів.

Програму було вирішено робити за допомогою PHPStorm та Open Server Panel. В Open Server буде створено домен майбутнього сайту, а в PHPStorm писатиметься програмне забезпечення веб-додатку.

Середовище Open Server Panel буде використовуватися для створення серверу, оскільки даний програмний продукт оснащений великою кількістю сервісів та можливостей для інтернет-програмування. Він має важливі складові, які будуть використовуватись при створенні проекту. Наприклад, підтримку різних версій PHP, у тому числі PHP 7, а також додаток PHPMyAdmin для роботи с базами даних.

В проекті будуть використані технології:

1. HTML5.
2. CSS3.
3. JavaScript.
4. Паттерн MVC.
5. PHP7.
6. MySQL.
7. JQuery.

Програмне забезпечення буде писатись на PHP, і для цього чудово підійде таке середовище, як PHPStorm. Це додаток від JetBrains, який має широкий вибір можливостей, які будуть використовуватись при написанні ПЗ. Він добре взаємодіє з усіма описаними вище технологіями.

## **1.2 Аналіз існуючих веб-сайтів за тематикою курсового проекту**

На даний момент мережа Інтернет наповнена найрізноманітнішими інтернет-магазинами, тому можна роздивитись особливості їх інтерфейсу та можливостей для більш конкретної постановки вимог до зовнішнього вигляду та функціоналу додатку, що створюється. Розглянемо ряд веб-додатків за темою «Література», які розміщені в Інтернеті.

Онлайн-магазин “Yakaboo” (Рис. 1.1) пропонує великий асортимент книжок. Інтерфейс сайту досить складний, він має три панелі для переходу по сторінками: розділи з жанрами, різні популярні книги та новинки, та панель навігації з інформацією щодо правил магазину.

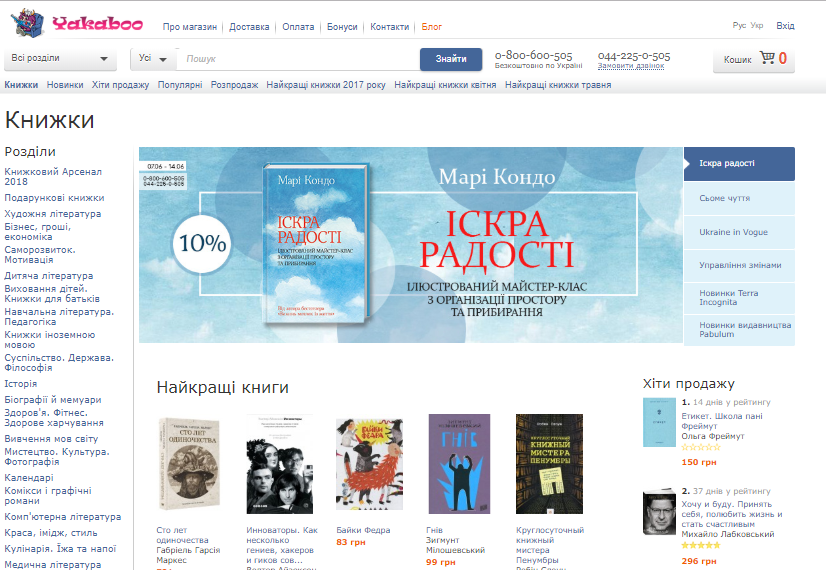


Рис. 1.1 – Інтернет-магазин «Yakaboo»

Такий інтерфейс важко сприймається та перенасичений інформацією. Також на головній сторінці винесено дуже багато одиниць товарів з кожного розділу, що також виглядає занадто яскраво.

Наступний сайт – «Читайка» (Рис. 1.2) має набагато приємніший інтерфейс. Контакти на головній сторінці, зручна панель навігації та невелика панель з навігацією за жанрами літератури. Такий інтерфес добре сприймається користувачем, але як і на попередньому веб-сайті, на головні сторінці викладено занадто багато позицій та інформації.

В обох розглянутих сайтах також є можливість реєстрації та авторизації користувачів.

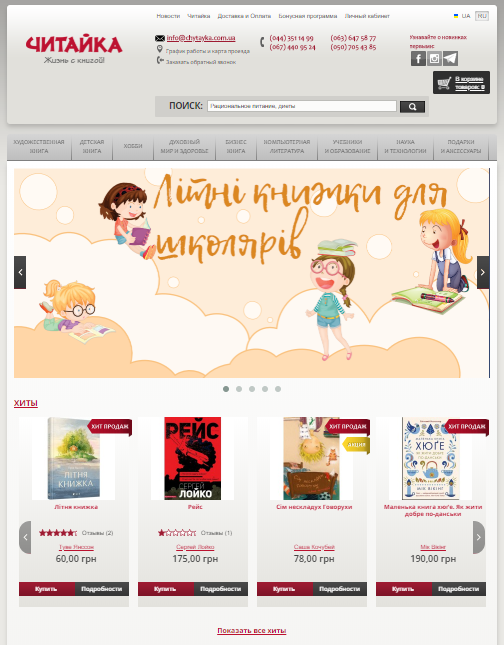
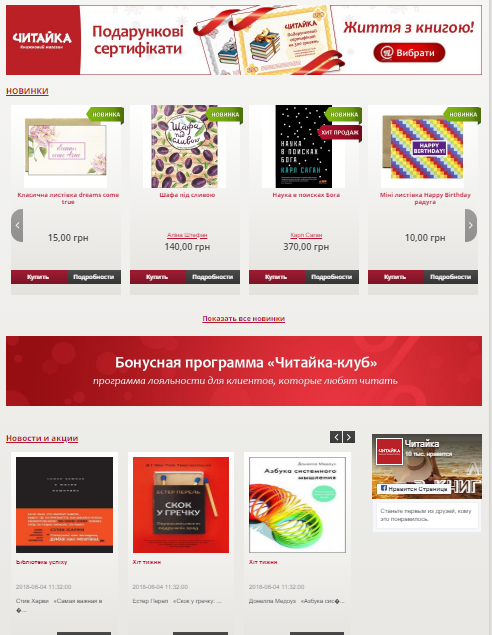
 

Рис. 1.2 – Головна сторінка веб-сайту «Читайка».

Ще один знайдений інтернет-магазин літератури – «ВсіКниги», що має простий інтерфейс, але досить складну орієнтацію по жанрам літератури, що вказано на Рис. 1.3.

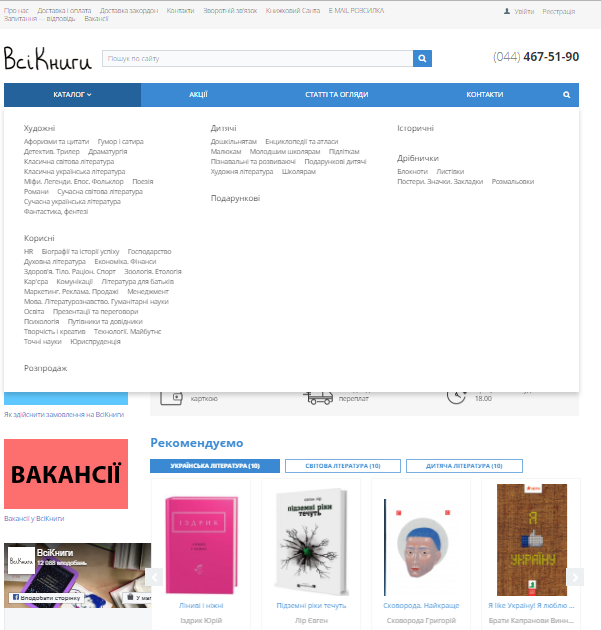


Рис 1.3 – Демонстрація навігації сайту «ВсіКниги»

Розглянувши існуючі веб-додатки за темою курсового проекта, можна сформувати основні характеристики програмного проекту, що створюється:

* Сайт повинен мати зручну панель навігації
* Має бути створений список жанрів літератури для переходу між розділами каталогу.
* Має бути створена сторінка реєстрації та авторизації користувачів.
* Сторінка каталогу не повинна буде містити занадто багато товарів, отже вона матиме розподілення на декілька секцій, які розгортатимуться за бажанням користувача.
* Має здійснюватися додавання товарів у кошик користувачем.
* Необхідно створити декілька сторінок з правилами сайту та контактами.

## **1.3 Технічне завдання до курсовий проект**

*Введення*

Розробка веб-додатку «Інтернет-магазин книг» є актуальним завданням, оскільки ВД на тематику онлайн продажів зараз користуються дуже великою популярністю як серед звичайних користувачів, так і підприємців.

*Найменування й область застосування*

Темою даного курсового проекту є веб-додаток «Інтернет-магазин книг». Розроблений програмний продукт призначений для роботи в інтернеті в якості онлайн-сервісу для купівлі через Інтернет.

*Підстава для розробки:*

Підставою для розробки проекту є завдання на курсовий проект.

*Призначення розробки:*

Веб-додаток призначений для індивідуального використання користувачами з метою проведення дозвілля, отримання інформації про літературу та її купівлі.

*Вимоги до програмного засобу:*

Веб-додаток повинен надавати можливість здійснювати дії на веб-сайті, переходити по сторінкам та реєструватись на ньому.

Веб-додаток повинен виконувати наступні функції з наступними вхідними даними:

1. Повинна бути наявна можливість керування сайтом за допомогою кнопок і посилань.
2. Додаток повинен забезпечити можливість адміністрування, тобто додавання та редагування даних.
3. Також повинна бути розроблена частина користувача, в якій можна буде додавати товари в кошик та переглядати сторінки.

Додаток повинен мати зручний інтерфейс, зрозумілу структуру та навігаці. При зміні сторінки має змінюватись основна частина, а «header» та «footer» залишатись статичними.

*Вимоги до надійності:*

Передбачити всі недоліки та можливі помилки у роботі веб-додатку. Користувач не має бачити тексту помилок у разі виникнення виключень.

*Стадії та етапи розробки*

Розробка програмного засобу включає наступні етапи:

1. Аналіз вимог
2. Розробка технічного завдання
3. Проектування алгоритму роботи ВД
4. Розробка ВД
5. Тестування ВД

## **Висновки до першого розділу**

В першому розділі визначено спосіб розробки ВД, його властивості, структура сайту та його функціонал.

Також було проаналізовано і визначено необхідні інструменти для реалізації задуманого проекту, а також деякі елементи його інтерфейсу та можливостей ВД, що розробляється.

# РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РОЗРОБКА ВЕБ-САЙТУ

## **2.1 Проектування загального алгоритму роботи веб-додатку**

Розглянемо загальний визначений алгоритм роботи ВД.

Після завантаження ВС буде відкриватись головна сторінка з загальною інформацією про інтернет-магазин. Згори буде статична панель навігації, за допомогою якої користувач зможе увійти в профіль чи зареєструватись, і вже після цього продивлюватись товари та отримати можливість їх купівлі. Також у правому верхньому кутку буде кнопка «кошика»(Рис. 2.1).

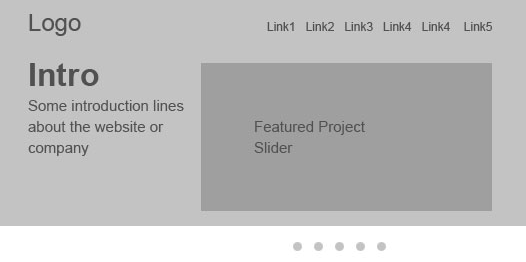


Рис. 2.1 – Приблизний макет головної сторінки

Не менш важливий модуль, який необхідно створити – каталог товарів. Ця сторінка поминна містити розподілення на категорії та товари. Кожен блок з товарною одиницею міститиме назву, зображення, автора книги та ціну. Макет сторінки можна створити за допомогою технологій Bootstrap(Рис. 2.2).

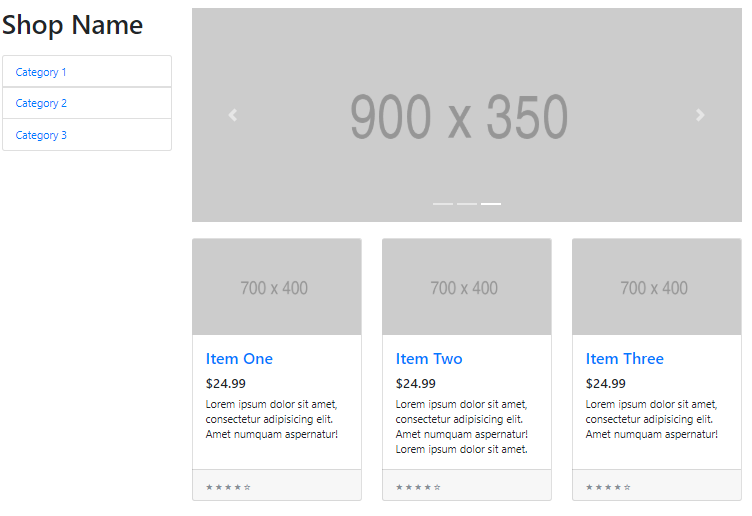


Рис. 2.2 – Приблизний макет каталогу

Також для кожного товару буде створена сторінка(Рис. 2.3), до якої з бази даних буде підгружатись вся необхідна інформація про продукт. Ця сторінка відкриватиметься при натисканні на товар в каталозі. На ній буде представлена більш обширна інформація: кількість сторінок в книзі, рік видання і т.д.

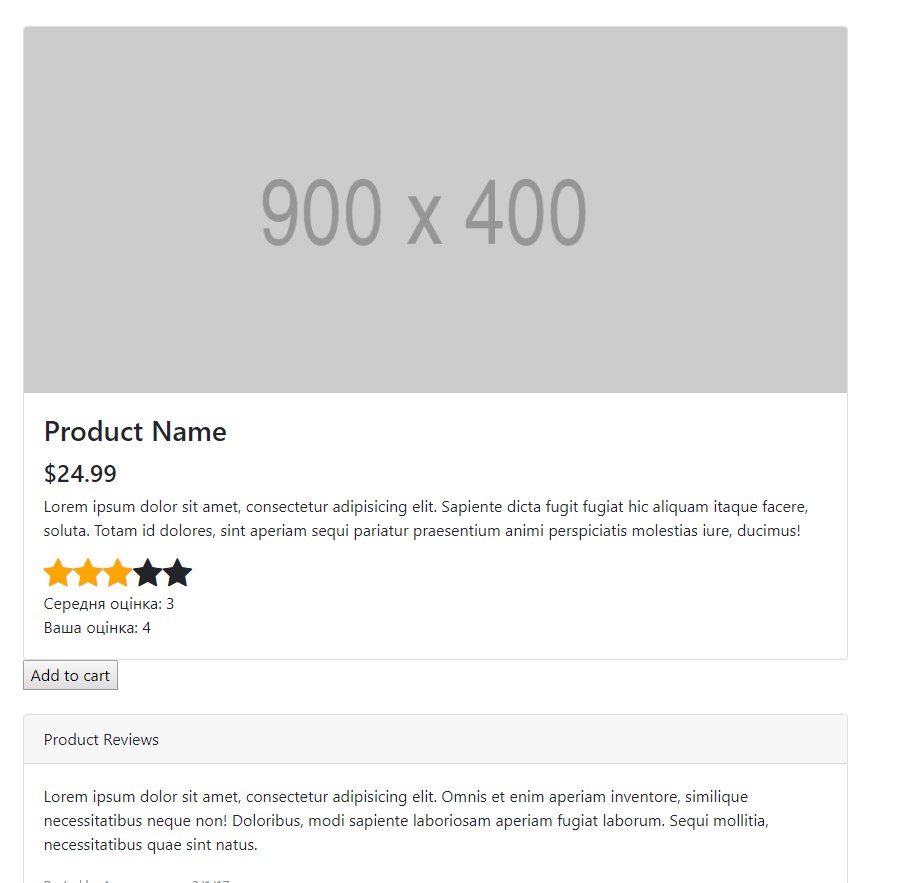


Рис. 2.3 – Сторінка перегляду одного товару

Створимо форми реєстрації та входу користувача на сайт(Рис. 2.4, 2.5). Це також необхідно для подальшої розробки ПЗ.

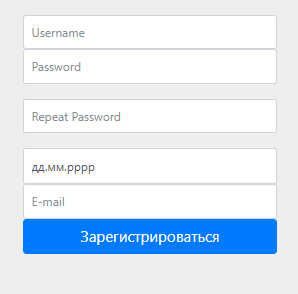
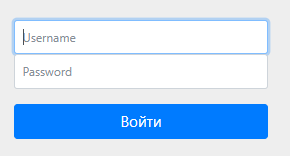
 

Рис. 2.4 – Форма реєстрації Рис. 2.5 – Форма входу користувача

користувача

Далі буде розроблено основний контролер, який відкриватиме необхідну сторінку за допомогою обробки запитів. Також пізніше буде розроблений власний дизайн ВД, дані макети створюються для розробки основної частини ПЗ.

## **2.2 Розробка функціональних алгоритмів роботи веб-сайту**

Розробка CMS-системи перш за все означає створення правильної структури файлів для можливості легко вносити зміни до системи адміністрування без утворення виключень та помилок.

Для коректної роботи додатку буде використано паттерн MVC. Після створення проекту в PHPStorm необхідно створити його структуру та розподілити файли на окремі директорії (Рис. 2.6).

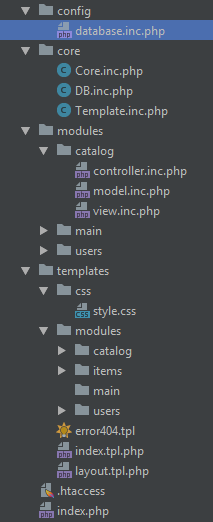


Рис. 2.6 – Початкова структура проекту

В папці «config» міститимуться файли для запису констант, які завжди будуть необхідні при розробці програмного коду додатку.

Директорія «core» містить в собі файли з основними методами для роботи з базою даних та ВС загалом.

Далі створюється папка модулів, в яких і буде використовуватись паттерн MVC. В кожному модулі буде створено три файли: файл-модель, файл-контролер та файл-представлення. Контролер отримуватиме дані від користувача, далі обробляти їх та викликати певні методи моделі для обробки цих даних. Після цього за допомогою view-частини користувач отримуватиме результат обробки поданих даних(наприклад, переходу на сторінку чи відправки форми). Саме файли модулів і будуть вагомою складовою при функціонуванні сайту.

У директорії «templates» розміщуються сторінки, створені за допомогою html, таблиці стилів, файли JS.

Також в кореневій папці проекту створюється файл .htacces, що містить загальні налаштування сторінки.

Файл index.php – основний обробник запитів користувачів. З нього починається розробка всього проекту.

Для зберігання даних використовуватиметься база даних MySQL. Для основних модулів створимо окремі таблиці(Рис. 2.7). Їх зміст зрозумілий за назвами таблиць.

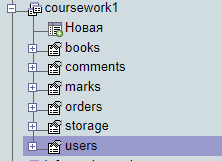


Рис. 2.7 – Структура бази даних веб-додатку

## **2.3** **Розробка веб-додатку**

У раніше створеному проекті можна розпочинати розробку основних контролерів. Для початку в файлі .htacces пропишемо основні налаштування сервера:

**AddDefaultCharset utf-8  
RewriteEngine on  
RewriteBase /  
RewriteCond %{REQUEST\_FILENAME} !-f  
RewriteCond %{REQUEST\_FILENAME} !-d  
RewriteRule ^(.\*)$ index.php?path=$1**

Головна мета наведеного коду – переадресація запитів на файл index.php.

Далі розробимо ПЗ для обробки запитів у ньому.

Лістинг:

**<?php  
include**(**'config/database.inc.php'**);  
*error\_reporting*(***E\_ALL***);  
*set\_include\_path*(**'core'**);  
*spl\_autoload\_extensions*(**'.inc.php'**);  
*spl\_autoload\_register*(**function** ($className){  
 $path = **'core/'**.$className.**'.inc.php'**;

**if**(*is\_file*($path))  
 **include**($path);  
});  
*spl\_autoload\_register*(**function**($className)  
{  
 $parts = *explode*(**'\_'**,$className);  
 $path = **'modules/'**.*strtolower*(*array\_shift*($parts)).  
 **'/'**.*strtolower*(*array\_shift*($parts)).**'.inc.php'**;  
 **if**(*is\_file*($path))  
 **include**($path);  
});  
*date\_default\_timezone\_set*(**'Europe/Kiev'**);  
Core::*Init*();  
Core::*Run*();  
Core::*Done*();

Даний код забезпечує формування рядку запиту для виклику відповідного контролера.

Цей файл посилаєтьс на файл Core.inc.php. Це файл класу, який містить наступні основні методи:

**public static function** Init()  
{  
 *session\_start*();  
 **self**::*$IndexTpl* = **new** Template(**'templates/layout.tpl.php'**);  
 **self**::*$IndexTpl*->setParam(**'Title'**, **''**);  
 **self**::*$IndexTpl*->setParam(**'Content'**,**'some content overhere'**);  
 **self**::*$IndexTpl*->setContentView(**'templates/index.tpl.php'**);  
 **self**::*$DB* = **new** DB(***DB\_HOST***,***DB\_USER***,***DB\_PASS***,***DB\_NAME***);  
 *// db connect*}  
  
**public static function** Run()  
{  
 **if**(**isset**($\_GET[**'path'**]) && !**empty**($\_GET[**'path'**])){  
 $path = $\_GET[**'path'**];  
 $pathParts = *explode*(**'/'**,$path);  
  
 $className = *ucfirst*(*array\_shift*($pathParts).**'\_Controller'**);  
 $methodShortName = *ucfirst*(*array\_shift*($pathParts));  
 **if**(**empty**($methodShortName))  
 $methodShortName = **'Index'**;  
 $methodName = $methodShortName.**'Action'**;  
 }  
 **else**{  
 $className = **'Main\_Controller'**;  
 $methodName = **'IndexAction'**;  
 $pathParts = **array**();  
 }  
  
  
 $error404 = **false**;  
 **if**(*class\_exists*($className)){  
 $controller = **new** $className();  
 **if**(*method\_exists*($controller,$methodName)){  
 $paramsArray = $controller->$methodName($pathParts);  
 **self**::*$IndexTpl*->setParams($paramsArray);  
 }**else** $error404 = **true**;  
 }**else** $error404 = **true**;  
  
 **if** ($error404)  
 {  
 $controller = **new** Main\_Controller();  
 $res = $controller->Error404Action();  
 }  
}  
  
**public static function** Done()  
{  
 **self**::*$IndexTpl*->display();  
}

Метод Init() відповідає за з’єднання з базою даних, Run() забезпечує виклик контролера та екшена, вказаних у рядку запиту, а Done() відображатиме отримані дані.

Повний код файлу Core.inc.php можна переглянути в Додатку А.

Створено основний макет з «шапкою» та «підвалом» сайту, в який буде завантажуватись контент різних сторінок.

Лістинг:

<**body id="page-top"**>  
  
*<!-- Navigation -->*<**nav class="navbar navbar-expand-lg navbar-dark bg-dark fixed-top" id="mainNav"**>  
 <**div class="container"**>  
 <**a class="navbar-brand js-scroll-trigger" href="/"**>Букіністъ</**a**>  
 <**button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarResponsive" aria-controls="navbarResponsive" aria-expanded="false" aria-label="Toggle navigation"**>  
 <**span class="navbar-toggler-icon"**></**span**>  
 </**button**>  
 <**div class="collapse navbar-collapse" id="navbarResponsive"**>  
 <**ul class="navbar-nav ml-auto"**>  
 <**a class="nav-link js-scroll-trigger" href='/catalog/list'**>Каталог</**a**>  
 **<?php  
 if**(**isset**($\_SESSION[**'login'**]))  
 {  
 **if**($\_SESSION[**'role'**]==**'admin'**)  
 {  
 **echo '<li class="nav-item">  
 <a class="nav-link js-scroll-trigger" href="/catalog/addBook">Додати книгу</a>  
 </li>'**;  
 **echo '<li class="nav-item">  
 <a class="nav-link js-scroll-trigger" href="/catalog/editBook">Змінити/видалити книгу</a>  
 </li>'**;  
 }  
 **else** {  
 **echo '  
 <li class="nav-item">  
 <a class="nav-link js-scroll-trigger" href="#about">About</a>  
 </li>'**;  
 }  
 **echo '<li class="nav-item"><a href="/users/profile" >**

**<span class="nav-link js-scroll-trigger" >Ви авторизовані як <span id="userlogin">'**.$\_SESSION[**'login'**].**'</span></span></a>  
 </li>'**;  
 **echo '<li class="nav-item">  
 <a class="nav-link js-scroll-trigger" href="/users/logout">Вийти</a>  
 </li>'**;  
 **if**(**isset**($\_COOKIE[**'order'**]))  
 **echo "<li class=\"nav-item\">  
 <a class=\"nav-link js-scroll-trigger\" href='/orders/cart'>Кошик ( <span>"**.*count*(*json\_decode*($\_COOKIE[**'order'**])).**"</span> )</a>  
 </li>"**;  
 }  
 **else** {  
 **echo '<li class="nav-item">  
 <a class="nav-link js-scroll-trigger" href="/users/login">Увійти</a>  
 </li>  
 <li class="nav-item">  
 <a class="nav-link js-scroll-trigger" href="/users/registration">Зареєструватись</a>  
 </li>'**;  
 }  
 **?>** </**ul**>  
 </**div**>  
 </**div**>  
</**nav**>  
  
**<?php include**($content\_view)**?>***<!-- Footer -->*<**footer class="py-5 bg-dark"**>  
 <**div class="container"**>  
 <**p class="m-0 text-center text-white"**>Copyright **&copy;** Vasylevska V 2018</**p**>  
 </**div**>  
 *<!-- /.container -->*</**footer**>  
</**body**>

Весь код сторінки можна переглянути в Додатку Б. У рядку «<?php include($content\_view)?>» і буде відбуватись підстановка контенту по запитам користувача.

Ддя створеного макету потрібно написати декілька варіацій контенту – головна сторінка і каталог, наприклад. Ці макети було розроблено в пункті 2.1, та їх лістинг можна переглянути у Додатку В. Створюємо відповідні модулі, що міститимуть три складові MVC. Розглянемо головну сторінку як приклад.

Перш за все розроблюємо контролер:

**<?php  
class** Main\_Controller  
{

**public function** IndexAction()  
 {  
 Main\_View::*indexView*();  
 }  
  
 **public function** Error404Action()  
 {  
 Core::*$IndexTpl*->setContentView(**'templates/error404.tpl'**);  
 }  
  
}

Цей контролер досить простий, і може викликати або файл перегляду головної сторінки, або сторінку помилки, якщо щось піде не так.

Файл view.inc.php:

**<?php  
class** Main\_View  
{  
 **public static function** indexView()  
 {  
 $tpl = Core::*$IndexTpl*;  
 $tpl->setContentView(**'templates/index.tpl.php'**);  
 }  
}

Він відправлятиме необхідні дані для перегляду у головний контролер, описаний раніше.

Також для подальшої роботи необхідно створити клас для роботи з базою даних. Це файл DB.inc.php, і він містить всі необхідні методи для занесення та зміни даних у БД. Наведемо найголовніші з них:

Лістинг з’єднання з БД:

**public function** \_\_construct($host,$user,$pass,$dbname)  
{  
 $this->**pdo** = **new** PDO(**"mysql:host=**{$host}**;dbname=**{$dbname}**"**,$user,$pass);  
}

Лістинг методу реєстрації та входу користувачів:

**public function** RegisterUser($row)  
{  
 $stmt = $this->**pdo**->prepare(**'insert into users(login,email,password,token,birthdate) values(:login,:email,:password,:token,:birthdate)'**);  
 $stmt->bindParam(**':login'**,$row[**'login'**]);  
 $stmt->bindParam(**':password'**,$row[**'password'**]);  
 $stmt->bindParam(**':birthdate'**,$row[**'bday'**]);  
 $stmt->bindParam(**':token'**,$row[**'token'**]);  
 $stmt->bindParam(**':email'**,$row[**'email'**]);  
 $stmt->execute();  
}  
  
**public function** LoginUser($login,$password)  
{  
 $stmt = $this->**pdo**->prepare(**'select** *count***(***\****) from users where login=:l and password=:p and status=1'**);  
 $stmt->bindParam(**':l'**,$login);  
 $stmt->bindParam(**':p'**,$password);  
 $stmt->execute();  
  
 $res = $stmt->fetch();  
 **return** $res[0];  
}

Лістинг методу для додавання товару:

**public function** AddBook($row)  
{  
 $stmt = $this->**pdo**->prepare(**'insert into books (author,name,price,genre,description,year, age, pages, language,imageId) values(:author,:name,:price,:genre,:description,:year, :age, :pages, :language, :imageId)'**);  
 $stmt->bindParam(**':name'**,$row[**'name'**]);  
 $stmt->bindParam(**':author'**,$row[**'author'**]);  
 $stmt->bindParam(**':price'**,$row[**'price'**]);  
 $stmt->bindParam(**':genre'**,$row[**'genres'**]);  
 $stmt->bindParam(**':description'**,$row[**'description'**]);  
 $stmt->bindParam(**':year'**,$row[**'year'**]);  
 $stmt->bindParam(**':age'**,$row[**'age'**]);  
 $stmt->bindParam(**':pages'**,$row[**'pages'**]);  
 $stmt->bindParam(**':language'**,$row[**'language'**]);  
 $stmt->bindParam(**':imageId'**,$row[**'imageId'**]);  
 $stmt->execute();  
  
  
 $stmt = $this->**pdo**->prepare(**'SELECT LAST\_INSERT\_ID() as id'**);  
 $stmt->execute();  
 $bookid = $stmt->fetchAll()[0][0];  
  
  
 $stmt = $this->**pdo**->prepare(**'insert into storage (bookid,amount) values (:bookid,:amount)'**);  
 $stmt->bindParam(**':bookid'**,$bookid);  
 $stmt->bindParam(**':amount'**,$row[**'amount'**]);  
 $stmt->execute();  
}

Повний лістинг файлу DB.inc.php можна переглянути у Додатку Г, оскільки він містить дуже велику кількість методів.

Також створено клас Template, в якому прописане коректне відображення отриманих даних та контенту. Його лістинг можна переглянути в Додатку Д.

Далі розробимо інші модулі зі схожими властивостями, перш за все – модуль каталогу.

Контролер каталогу міститиме особливі методи, які викликатимуть певні частини коду у моделі, а звідти відповідно і в самих класах, створених на початку. Приклади:

* Видалення товару

**public function** DeleteBookAction()  
{  
 Core::*SetIsAjaxRequest*();

Catalog\_Model::*DeleteBook*($\_POST[**'id'**]);  
}

* Зміна характеристик товару

**public function** UpdateBookAction()  
{  
 Catalog\_Model::*UpdateBook*($\_POST,$\_FILES);  
  
 *header*(**'Location: /catalog/editBook'**);  
}

Файл містить і інші функції, лістинг модулю Каталогу можна переглянути в Додатку Е.

Наведемо приклади методів моделей, які відповідають викликаним вище у контролері.

Видалення:

**public static function** DeleteBook($id)  
{  
 $db = Core::*GetDB*();  
  
 $res = $db->DeleteBook($id);  
  
}

Редагування:

**public static function** UpdateBook($row,$files)  
{  
 **if**(**isset**($files[**'img'**]) && $files[**'img'**][**'size'**]!=0)  
 {  
 $poster = $files[**'img'**];  
 $row[**'imageId'**] = *uniqid*().**'.'**.*explode*(**'/'**,$poster[**'type'**])[1];  
 $target = **'uploads/books/'**.$row[**'imageId'**];  
 *move\_uploaded\_file*($poster[**'tmp\_name'**],$target);  
 }  
 $db = Core::*GetDB*();  
  
 $res = $db->UpdateBook($row);  
}

Тобто, файли моделі викликають раніше створені методи у класі DB.

За аналогією створюємо і інші файли за утвердженим шаблоном MVC. Окрім вже перерахованих, створені наступні модулі: користувача, замовлень, коментарів та оцінювання товарів. Всього було створено 6 модулів, на враховуючі окремі сторінки для адміністрування (Рис. 2.8). Кожен з модулів є спільні методи, схожі між собою: занесення вказаних даних в БД, відображення даних у коректному вигляді та обробка дій користувача. Розписувати їх буде недоречно, тому детально переглянути лістинг модулів та деяих файлів можна буде у додатках.

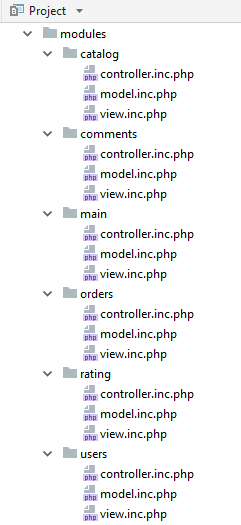


Рис. 2.8 – Структура папки модулів

Також розроблений власний дизайн інтернет-магазину, який наведено в додатку.

## **Висновки до другого розділу**

В даному розділі курсового проектування було проведено розробку загальної схеми роботи програми, створеної на основі проаналізованих питань.

Був розроблений проект в PHPStorm, а в ньому досить складна стуктура файлів. На етапі розробки першого рівня були створені необхідні макети для подальшого створення ПЗ сайту. Була створена велика кількість модулів з MVC файлами, щоб зробити програмну частину більш розподіленою. Створені сторінки входу та реєстрації користувачів, каталогу, кошику та головну, а також модулі коментарів, оцінювання. Були реалізовані методи роботи з запитами та базою даних для збереження інформації, вказаної адміністратором або звичайним користувачем.

В результаті було отримано готовий програмний продукт з потрібною функціональністю, який виконує поставлені задачі та вимоги.

# РОЗДІЛ 3. ОПИС РОБОТИ З ВЕБ-САЙТОМ ТА ЙОГО ТЕСТУВАННЯ

## **Опис роботи з веб-сайтом**

Даний розділ курсового проекту передбачає наявність готового веб-додатку і необхідність пояснення його роботи для користувача, а також його тестування. Пояснення відбуватиметься за допомогою графічних зображень та текстових роз’яснень необхідних дій.

При відкритті посилання веб-додатку користувачу відкриється головна сторінка, яка містить основну інформацію про інтернет-магазин та статичну панель навігації. Зображення сторінки можна побачити на рисунку 3.1 та 3.2.

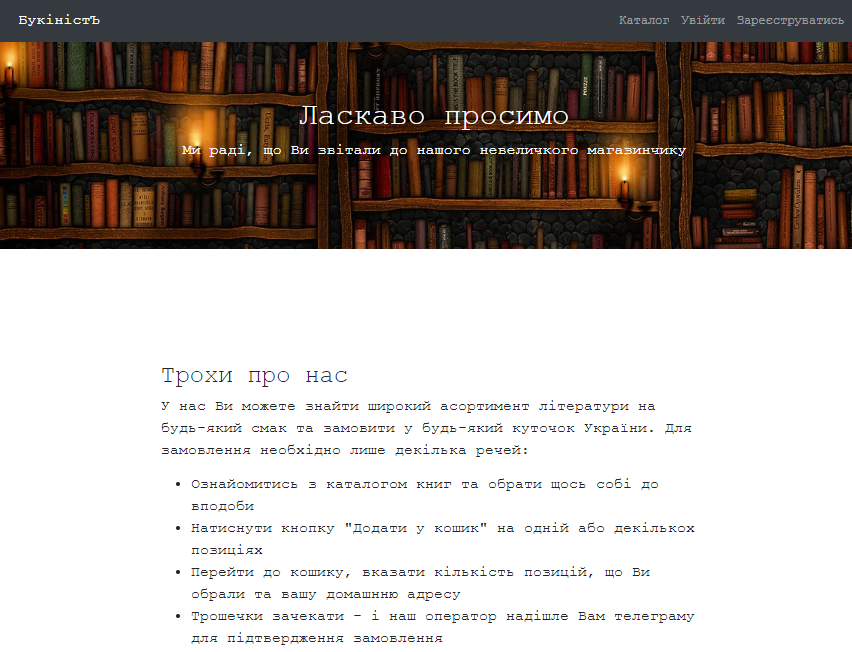


Рис. 3.1 – Вигляд головної сторінки (перша частина)



Рис. 3.2 – Вигляд головної сторінки (друга частина)

На головній сторінці користувач матиме можливість ознайомитись з представленою інформацією та перейти до каталогу, реєстрації або входу в особистий кабінет, з якого він вже матиме можливість замовляти представлені товари.

Перейдемо до сторінки реєстрації(Рис. 3.3)

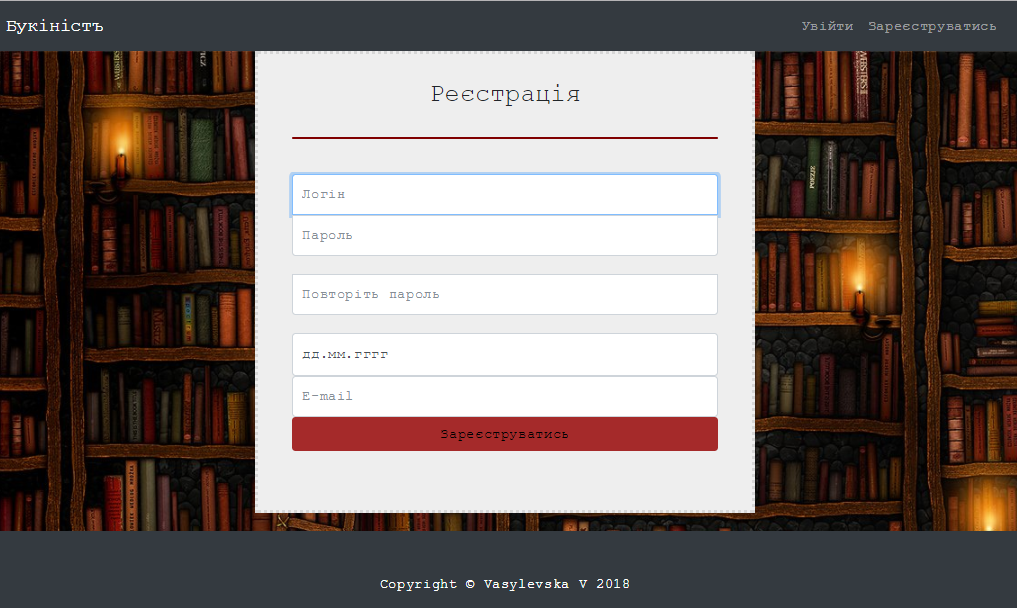


Рис. 3.3 – Форма реєстрації користувача

На цій сторінці користувач має заповнити поля даних про себе. Логін та пошта мають бути унікальними. При вдалій реєстрації він має підтвердити реєстрацію на поштовій скринці, при невдалій – отримає повідомлення про це.

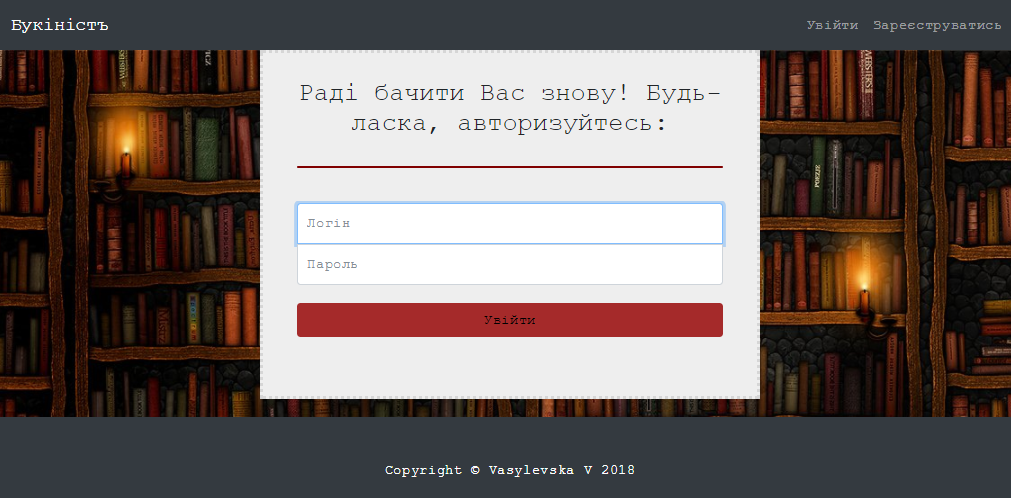


Рис. 3.4 – Форма входу користувача

При правильному введенні логіну та пароля в залежності від ролі (покупець чи адміністратор), юзери отримають трохи різне представлення сторінок. Розглянемо частину і адміністратора, і покупця(Рис. 3.5 та 3.6 відповідно).

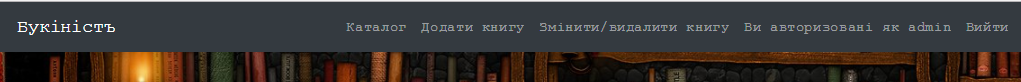


Рис. 3.5 – Вигляд панелі навігації для адміністратора

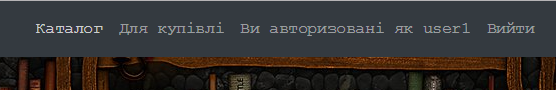


Рис. 3.6 – Вигляд панелі навігації для звичайного користувача

Як видно з панелі, адміністратор як і користувач має можливість продивлятись товари.

Роздивимося окремо модулі адміністратора.

Адміністратор має можливість додавати товари та змінювати їх кількість на сайті, перейшовши у розділ «Додати книгу» (Рис. 3.7).

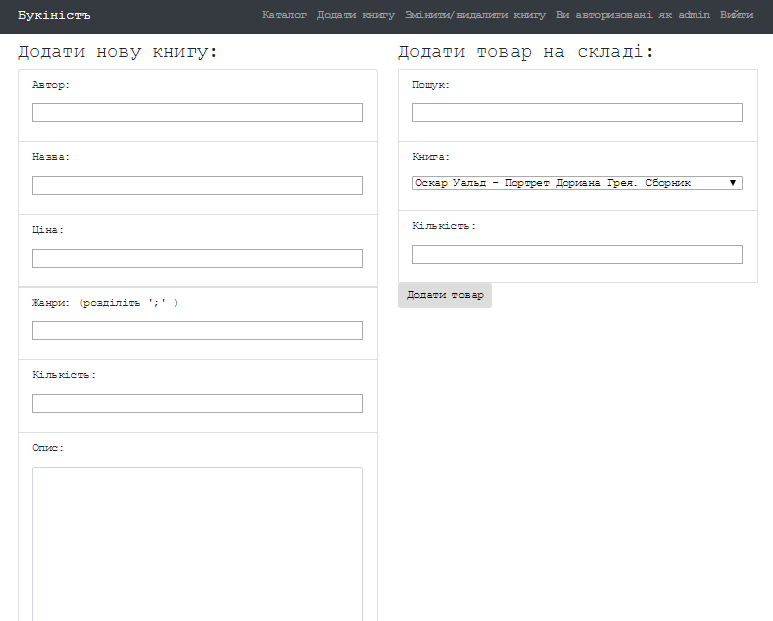


Рис. 3.7 – Форми для додавання книг та їх кількості в наявності

Перша форма містить всі поля, що є і в БД з товарами. Таким чином адміністратор може створювати одиниці товару, заповнивши та відправивши форму.

Друга форма служить для зміни кількості товарів на складі, якщо відбулась нова поставка, наприклад. Для зручності у формі передбачено поле пошуку.

Також створена сторінка редагування та видалення товарів(Рис. 3.8). На ній також є поле пошуку. На фрму автоматично підтягуються дані з БД і адміністратор має можливість вносити зміни до них, або зовсім видалити позицію.

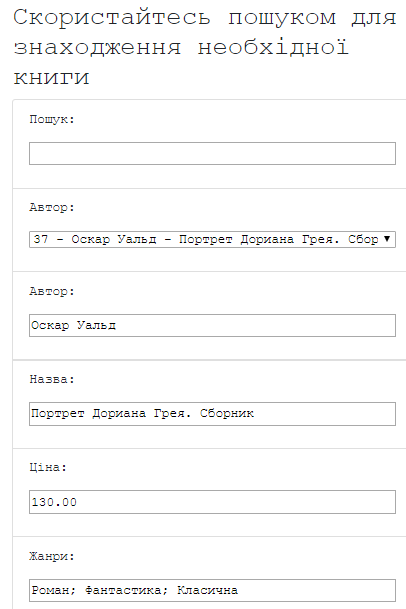


Рис. 3.8 – Форма редагування товару

Тепер перейдемо до загальних модулів. Перш за все, це каталог товарів, який стає доступним піля авторизації на сайті(Рис. 3.9).

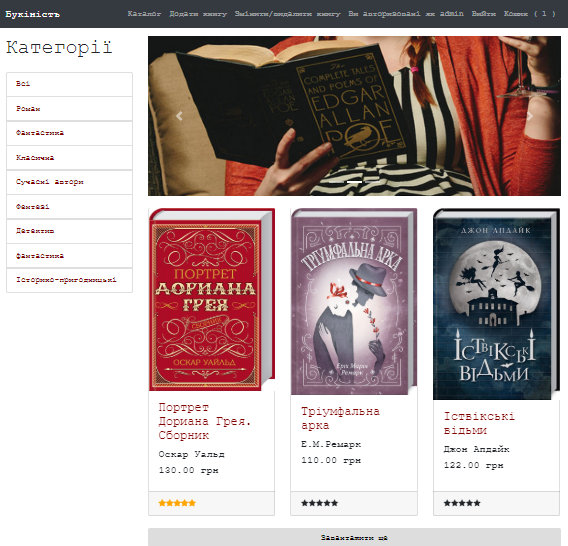


Рис. 3.9. – Каталог товарів

Тут користувач може перемикати жанри, за якими сортуються товари. На сторінку викладається спочатку три товари, а після натискання кнопки «Завантажити ще», ще три, і так далі, доки товари в даній категорії не закінчаться. Щоб придбати книгу, користувач має натиснути на одну з них і відкриється сторінка з одним товаром(Рис. 3.10).

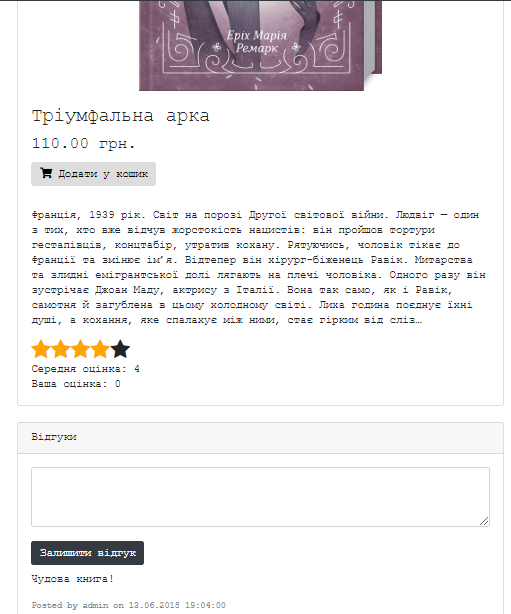


Рис. 3.10 – Сторінка з однією книгою

На цій сторінці користувач може залишити відгук, оцінити товар та додати його до корзини будь-яку кількість раз. Кнопка кошику з’являється тільки після додавання якогось товару до неї(Рис. 3.11).

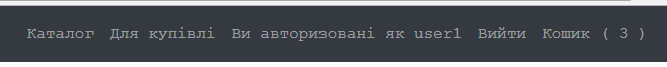


Рис. 3.11 – Утворення кнопки кошика після додавання у нього товару

У самому кошику вказується обраний товар та його короткі характеристики. Можна змінити його кількість, вказати адресу доставки та здійснити замовлення(Рис 3.12).

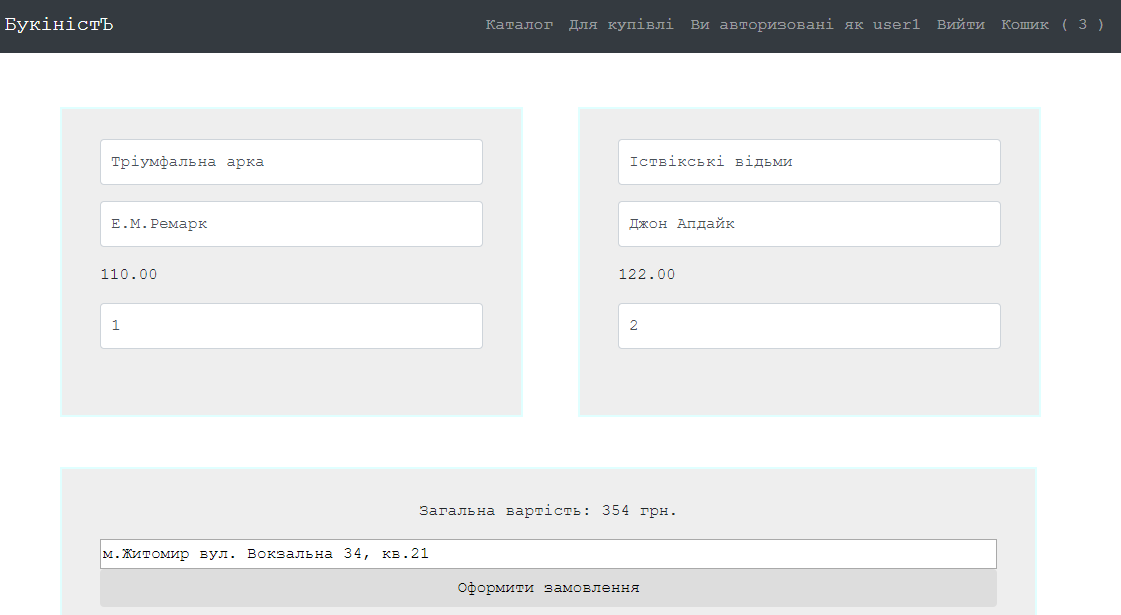


Рис. 3.12 – Вигляд кошику

При натисканні на «Ви авторизовані як …» користувачу відкриється його власний кабінет, де будуть вказані всі його замовлення(Рис. 3.13).

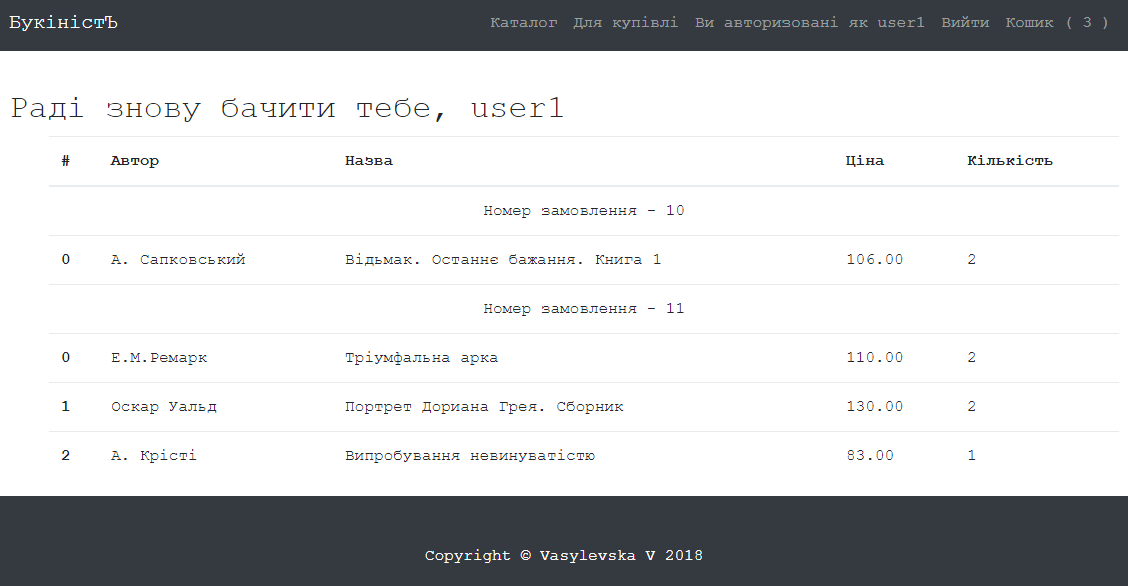


Рис. 3.13 – Особистий кабінет користувача

Кнопка «Вийти» повертає користувача у режим простого перегляду без доступу до каталогу та замовлення продуції.

Також при введенні користувачем неіснуючого шляху буде відкриватись сторінка 404(Рис. 3.14).

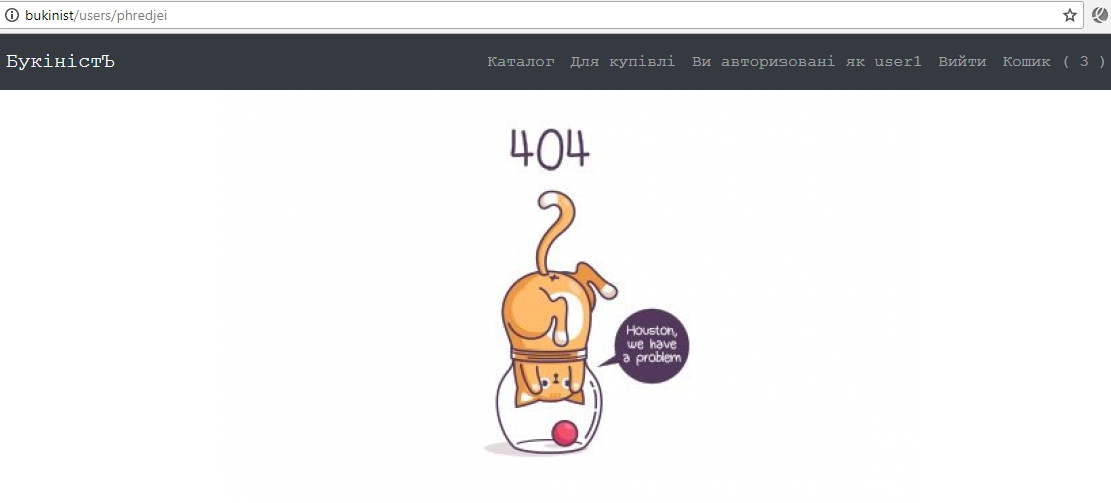


Рис. 3.14 – Сторінка помилки

## **3.2 Тестування веб-додатку**

Був проведений ряд тестувань, перш за все альфа-тестування розробником. Виникала достатня кількість помилок при розробці веб-додатку, але після закінчення його написання кількість помилок знизилась до 0. Були проаналізовані функції і додані деякі нові можливості.

Заради безпеки даних користувача пароль почав переводитись за допомогою функції md5(). Так навіть розробник не дізнається пароль користувача(Рис 3.15).

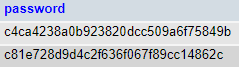


Рис. 3.15 – Вигляд паролів після кодування

Також не відразу було додано підтвердження реєстрації через поштову скриньку. Ця функція була написана також з метою покращення безпеки даних користувачів. Вона викликала деякі проблеми при написанні, але всі вони були усунені, оскільки OS пропонує сервіс навіть для e-mail розсилки.

Неоднократно змінювався зовнішній вигляд веб-додатку та додавались нові дані. Спочатку товари не мали великої кількості характеристик, але згодом вони були додані для поширення інформації про ассортимент магазину.

Тестування показало, що помилки та виключення були наявні на ранішніх стадіях розробки ВД. Після створення наступних наступних сторінок, модулів та файлів кількість неточностей зменшувалась, як і кількість отримання виключень.

## **Висновки до третього розділу**

В даному розділі був досліджений функціонал створеного веб-додатку, а також розписані правила користування ним. Наведені приклади роботи веб-сайту та проаналізована його працездатність.

Опис роботи з додатком включає в себе роз’яснення із зображеннями сайту, які пояснюють принципи користування веб-додатком та опис елементів інтерфейсу для його користувачів.

При проведенні тестування додаток було запущено на виконання у реальних умовах, при яких він добре себе показав. Помилки було з легкістю усунено, і програмний продукт став повністю готовим для користування.

# ВИСНОВКИ

В результаті виконання курсового проекту отримано веб-додаток, який по функціональності повністю відповідає поставленим на початку проектування вимогам. Веб-додаток є простим у користуванні з інтуїтивно зрозумілим інтерфейсом.

В першому розділі курсового проекту було проаналізовано існуючі веб-програми схожого функціоналу та сформовано уявлення про веб-додаток, що розробляється. Також було визначено необхідні засоби та структурні елементи, що необхідні для реалізації додатку.

В другому розділі було проведено розробку загального і функціонального алгоритмів веб-додатку. При розробці функціонального алгоритму розроблено уявлення про методи розробки програмного коду додатку. На заключному етапі розділу було отримано готовий веб-додаток з потрібною функціональністю, який виконує всі поставлені задачі та вимоги.

В третьому розділі було описано методику користування створеним веб-додатком та проведено тестування створеного сайту, тобто було запущено на виконання у реальних умовах. В ході тестування було показано приклад створених покращень для забезпечення нормальної роботи додатку, та доказано, що створений програмний продукт є повністю готовим до використання.

Програмне забезпечення представляє собою папку з файлами, що утворились при розробці проекту. Для перегляду сайту його потрібно запускати через OpenServer, оскільки отриманий додаток має досить складну структуру.

Спроектований веб-додаток повністю відповідає поставленим вимогам та є достатньо швидкодіючим, а також не займає великої кількості ресурсів комп’ютера, на якому він запущений.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Как сверстать веб-страницу [Електронний ресурс] – Режим доступу: <https://habr.com/post/202408/>
2. Как сверстать веб-страницу. Часть 2 — Bootstrap [Електронний ресурс] – Режим доступу: <https://habr.com/post/211032/>
3. Строим свою CMS на PHP и MySQL [Електронний ресурс] – Режим доступу: <https://ruseller.com/project.php?id=11>
4. Паттерн MVC и PHP, часть 1 [Електронний ресурс] – Режим доступу: <http://ashep.org/2013/mvc-pattern-php-1/#.Wx2vzoozbIV>
5. Введение в MVC, онлайн урок [Електронний ресурс] – Режим доступу: <https://devionity.com/ru/courses/creating-mvc-framework/introduction-to-mvc>
6. Практика разработки CMS [Електронний ресурс] – Режим доступу: <https://habr.com/post/210264/>
7. PHP PDO. Краткое руководство [Електронний ресурс] – Режим доступу: <http://dandreev.com/blog/php-pdo-kratkoe-rukovodstvo/>
8. Как работать с PDO? Полное руководство. [Електронний ресурс] – Режим доступу: <http://phpfaq.ru/pdo>
9. PHP и ООП [Електронний ресурс] – Режим доступу: <http://www.php.su/learnphp/phpoo/>
10. PHP, PDO. Примеры работы [Електронний ресурс] – Режим доступу: <http://make2it.blogspot.com/2014/03/php-pdo.html>

ДОДАТКИ

Додаток А

Core.inc.php

**<?php  
  
class** Core  
{  
 **public static** *$IndexTpl*;  
 **private static** *$DB*;  
  
 **public static function** SetMainContent($file)  
 {  
 **self**::*$IndexTpl*->setContentView($file);  
 }  
  
 **public static function** SetIsAjaxRequest()  
 {  
 **self**::*$IndexTpl*->AjaxRequest();  
 }  
  
 **public static function** GetDB()  
 {  
 **return self**::*$DB*;  
 }  
  
 **public static function** Init()  
 {  
 *session\_start*();  
 **self**::*$IndexTpl* = **new** Template(**'templates/layout.tpl.php'**);  
 **self**::*$IndexTpl*->setParam(**'Title'**, **''**);  
 **self**::*$IndexTpl*->setParam(**'Content'**,**'some content overhere'**);  
 **self**::*$IndexTpl*->setContentView(**'templates/index.tpl.php'**);  
 **self**::*$DB* = **new** DB(***DB\_HOST***,***DB\_USER***,***DB\_PASS***,***DB\_NAME***);  
 *// db connect* }  
  
 **public static function** Run()  
 {  
 **if**(**isset**($\_GET[**'path'**]) && !**empty**($\_GET[**'path'**])){  
 $path = $\_GET[**'path'**];  
 $pathParts = *explode*(**'/'**,$path);  
  
 $className = *ucfirst*(*array\_shift*($pathParts).**'\_Controller'**);  
 $methodShortName = *ucfirst*(*array\_shift*($pathParts));  
 **if**(**empty**($methodShortName))  
 $methodShortName = **'Index'**;  
 $methodName = $methodShortName.**'Action'**;  
 }  
 **else**{  
 $className = **'Main\_Controller'**;  
 $methodName = **'IndexAction'**;  
 $pathParts = **array**();  
 }  
  
  
 $error404 = **false**;  
 **if**(*class\_exists*($className)){  
 $controller = **new** $className();  
 **if**(*method\_exists*($controller,$methodName)){  
 $paramsArray = $controller->$methodName($pathParts);  
 **self**::*$IndexTpl*->setParams($paramsArray);  
 }**else** $error404 = **true**;  
 }**else** $error404 = **true**;  
  
 **if** ($error404)  
 {  
 $controller = **new** Main\_Controller();  
 $res = $controller->Error404Action();  
 }  
 }  
  
 **public static function** Done()  
 {  
 **self**::*$IndexTpl*->display();  
 }  
}

Додаток Б

layout.tpl.php

<!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=""**>  
  
 <**title**>БукіністЪ</**title**>  
  
 *<!-- Bootstrap core CSS -->*<**link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.0.13/css/all.css" integrity="sha384-DNOHZ68U8hZfKXOrtjWvjxusGo9WQnrNx2sqG0tfsghAvtVlRW3tvkXWZh58N9jp" crossorigin="anonymous"**>  
 <**link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.2/css/bootstrap.min.css" integrity="sha384-PsH8R72JQ3SOdhVi3uxftmaW6Vc51MKb0q5P2rRUpPvrszuE4W1povHYgTpBfshb" crossorigin="anonymous"**>  
 <**script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"**></**script**>  
 <**script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.3/umd/popper.min.js" integrity="sha384-vFJXuSJphROIrBnz7yo7oB41mKfc8JzQZiCq4NCceLEaO4IHwicKwpJf9c9IpFgh" crossorigin="anonymous"**></**script**>  
 <**script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.2/js/bootstrap.min.js" integrity="sha384-alpBpkh1PFOepccYVYDB4do5UnbKysX5WZXm3XxPqe5iKTfUKjNkCk9SaVuEZflJ" crossorigin="anonymous"**></**script**>  
 <**script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"**></**script**>  
 <**link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css"**>  
 <**link href="/templates/css/style.css" rel="stylesheet" type="text/css"**>  
</**head**>  
  
<**body id="page-top"**>  
  
*<!-- Navigation -->*<**nav class="navbar navbar-expand-lg navbar-dark bg-dark fixed-top" id="mainNav"**>  
 <**div class="container"**>  
 <**a class="navbar-brand js-scroll-trigger" href="/"**>БукіністЪ</**a**>  
 <**button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarResponsive" aria-controls="navbarResponsive" aria-expanded="false" aria-label="Toggle navigation"**>  
 <**span class="navbar-toggler-icon"**></**span**>  
 </**button**>  
 <**div class="collapse navbar-collapse" id="navbarResponsive"**>  
 <**ul class="navbar-nav ml-auto"**>  
 <**a class="nav-link js-scroll-trigger" href='/catalog/list'**>Каталог</**a**>  
  
 **<?php  
  
 if**(**isset**($\_SESSION[**'login'**]))  
 {  
 **if**($\_SESSION[**'role'**]==**'admin'**)  
 {  
 **echo '<li class="nav-item">  
 <a class="nav-link js-scroll-trigger" href="/catalog/addBook">Додати книгу</a>  
 </li>'**;  
 **echo '<li class="nav-item">  
 <a class="nav-link js-scroll-trigger" href="/catalog/editBook">Змінити/видалити книгу</a>  
 </li>'**;  
 }  
  
 **echo '<li class="nav-item"><a href="/users/profile" >  
 <span class="nav-link js-scroll-trigger" >Ви авторизовані як <span id="userlogin">'**.$\_SESSION[**'login'**].**'</span></span></a>  
 </li>'**;  
 **echo '<li class="nav-item">  
 <a class="nav-link js-scroll-trigger" href="/users/logout">Вийти</a>  
 </li>'**;  
 **if**(**isset**($\_COOKIE[**'order'**]))  
 **echo "<li class=\"nav-item\">  
 <a class=\"nav-link js-scroll-trigger\" href='/orders/cart'>Кошик ( <span>"**.*count*(*json\_decode*($\_COOKIE[**'order'**])).**"</span> )</a>  
 </li>"**;  
 }  
 **else** {  
 **echo '<li class="nav-item">  
 <a class="nav-link js-scroll-trigger" href="/users/login">Увійти</a>  
 </li>  
 <li class="nav-item">  
 <a class="nav-link js-scroll-trigger" href="/users/registration">Зареєструватись</a>  
 </li>'**;  
 }  
  
 **?>** </**ul**>  
 </**div**>  
 </**div**>  
</**nav**>  
  
**<?php include**($content\_view)**?>***<!-- Footer -->*<**footer class="py-5 bg-dark"**>  
 <**div class="container"**>  
 <**p class="m-0 text-center text-white"**>Copyright **&copy;** Vasylevska 2018</**p**>  
 </**div**>  
 *<!-- /.container -->*</**footer**>  
  
  
<**script src="/templates/modules/catalog/orderController.js"**></**script**>  
  
</**body**>  
  
</**html**>

Додаток В

**Макети сторінок**

Лістинг розмітки головної сторінки:

<**header class="text-white bglib margtop"**>  
 <**div class="container text-center "**>  
 <**h1**>Ласкаво просимо</**h1**>  
 <**p class="lead"**>Ми раді, що Ви звітали до нашого невеличкого магазинчику</**p**>  
 </**div**>  
</**header**>  
<**section id="about"**>  
 <**div class="container"**>  
 <**div class="row"**>  
 <**div class="col-lg-8 mx-auto"**>  
 <**h2**>Трохи про нас</**h2**>  
 <**p class="lead"**>  
 У нас Ви можете знайти широкий асортимент літератури на будь-який смак та замовити у будь-який куточок України. Для замовлення необхідно лише декілька речей:  
 </**p**>  
 <**ul class="lead"**>  
 <**li**>Ознайомитись з каталогом книг та обрати щось собі до вподоби</**li**>  
 <**li**>Натиснути кнопку "Додати у кошик" на одній або декількох позиціях</**li**>  
 <**li**>Перейти до кошику, вказати кількість позицій, що Ви обрали та вашу домашнню адресу</**li**>  
 <**li**>Трошечки зачекати - і наш оператор надішле Вам телеграму для підтвердження замовлення</**li**>  
 </**ul**>  
 </**div**>  
 </**div**>  
 </**div**>  
</**section**>  
  
<**section id="services" class="bg-light"**>  
 <**div class="container"**>  
 <**div class="row"**>  
 <**div class="col-lg-8 mx-auto"**>  
 <**h2**>А ще Ви можете знайти нас у реальному світі!</**h2**>  
 <**p class="lead"**>  
 Ми знаходимось у Житомирі, за адресою вулиця Пушкінська, 9А.  
 Завітайте до нас, обирайте книжки та користуйтесь читальною залою скільки завгодно!  
 <**div class="col-lg-8"**>  
 <**img src="/uploads/bg/magazin.jpg" style="margin-left**: -5%**"**>  
 </**div**>  
  
 </**p**>  
 </**div**>  
 </**div**>  
 </**div**>  
</**section**>  
<**section id="contact"**>  
 <**div class="container"**>  
 <**div class="row"**>  
 <**div class="col-lg-8 mx-auto"**>  
 <**h2**>Поспілкуйтесь з нами!</**h2**>

<**p class="lead"**>  
 Будь-які питання Ви можете задати за телефоном ~380631010451<**br**>  
 Або в нашій <**a href=""** >групі в фейсбуці</**a**>  
 </**p**>  
 </**div**>  
 </**div**>  
 </**div**>  
</**section**>

Лістинг сторінки каталогу:

*<!-- Page Content -->*<**br**>  
<**br**>  
<**br**>  
<**div class="container"**>  
  
 <**div class="row"**>  
  
 <**div class="col-lg-3"**>  
  
 <**h1 class="my-4"**>Категорії</**h1**>  
 <**div class="list-group"**>  
 **<?php  
 echo '<label class="list-group-item form-check-label">  
 <a href="#" class="genre" data-genre="">Всі</a>  
 </label>'**;  
 **foreach** ($genres **as** $key => $item) {  
 **echo '<label class="list-group-item form-check-label">  
 <a href="#" class="genre" data-genre="'**.$item.**'">'**.$item.**'</a>  
 </label>'**;  
 }  
  
  
 **?>** *<!--  
 <a href="#" class="list-group-item">Category 1</a>  
 <a href="#" class="list-group-item">Category 2</a>  
 <a href="#" class="list-group-item">Category 3</a>-->* </**div**>  
  
 </**div**>  
 *<!-- /.col-lg-3 -->* <**div class="col-lg-9" id="items"**>  
  
 <**div id="carouselExampleIndicators" class="carousel slide my-4" data-ride="carousel"**>  
 <**ol class="carousel-indicators"**>  
 <**li data-target="#carouselExampleIndicators" data-slide-to="0" class="active"**></**li**>  
 <**li data-target="#carouselExampleIndicators" data-slide-to="1"**></**li**>  
 <**li data-target="#carouselExampleIndicators" data-slide-to="2"**></**li**>  
 </**ol**>  
 <**div class="carousel-inner" role="listbox"**>  
 <**div class="carousel-item active"**>  
 <**img class="d-block img-fluid" src="/uploads/bg/slide3.png" alt="First slide"**>  
 </**div**>  
 <**div class="carousel-item"**>  
 <**img class="d-block img-fluid" src="/uploads/bg/slide2.jpg" alt="Second slide"**>  
 </**div**>  
 <**div class="carousel-item"**>  
 <**img class="d-block img-fluid" src="/uploads/bg/slide1.jpeg" alt="Third slide"**>  
 </**div**>  
 </**div**>  
 <**a class="carousel-control-prev" href="#carouselExampleIndicators" role="button" data-slide="prev"**>  
 <**span class="carousel-control-prev-icon" aria-hidden="true"**></**span**>  
 <**span class="sr-only"**>Попередня</**span**>  
 </**a**>  
 <**a class="carousel-control-next" href="#carouselExampleIndicators" role="button" data-slide="next"**>  
 <**span class="carousel-control-next-icon" aria-hidden="true"**></**span**>  
 <**span class="sr-only"**>Наступна</**span**>  
 </**a**>  
 </**div**>  
  
 <**div class="row"**>  
 </**div**>  
 <**input type="button" id='loadmore' value="load more"**>  
  
 *<!-- /.row -->* </**div**>  
 *<!-- /.col-lg-9 -->* </**div**>  
 *<!-- /.row -->*</**div**>  
*<!-- /.container -->*<**script src="/templates/modules/catalog/bookLoader.js"**></**script**>

Лістинг сторінки з одним товаром:

*<!-- Page Content -->***<?php** *extract*($bookInfo)**?>**<**div class="container"**>  
 <**div class="row"**>  
 *<!--  
  
 <h1>***<?=**$id**?>***</h1>  
 <h1>***<?=**$name**?>***</h1>  
 -->* <**input type="hidden" id="bookid" value="<?=**$id**?>"**>  
 <**div class="col-lg-2"**>  
  
 </**div**>  
 *<!-- /.col-lg-3 -->* <**div class="col-lg-8"**>  
  
 <**div class="card mt-4"**>  
 <**img class="mainimg adminform" src="/uploads/books/<?=**$imageId**?>" alt=""**>  
  
 <**div class="card-body"**>  
 <**h3 class="card-title"**>**<?=**$name**?>**</**h3**>  
 <**div**>  
 <**h4**>**<?=**$price**?>** грн.</**h4**>  
 <**button type="button" value="<?=**$id**?>" class="btn btn-default btn-md addtocart"** >  
 <**span class="fas fa-shopping-cart"**></**span**> Додати в кошик  
 </**button**>  
 </**div**>  
 <**br**>  
 <**table class="table"**>  
 <**tr**>  
 <**td**>Рік видання</**td**>  
 <**td**>**<?=**$year**?>**</**td**>  
 </**tr**>  
 <**tr**>  
 <**td**>Кількість сторінок</**td**>  
 <**td**>**<?=**$pages**?>**</**td**>  
 </**tr**>  
 <**tr**>  
 <**td**>Мова</**td**>  
 <**td**>**<?=**$language**?>**</**td**>  
 </**tr**>  
 </**table**>  
 <**p class="card-text"**>**<?=**$description**?>**</**p**>  
  
 **<?php  
 require "templates/modules/rating/stars.tpl.php"**;  
 **?>** </**div**>  
 </**div**>  
 *<!-- /.card -->* <**div class="card card-outline-secondary my-4"**>  
 <**div class="card-header"**>  
 Product Reviews  
 </**div**>  
  
 <**div class="card-body" id="comment-container"**>  
 <**div**>  
 **<?php  
 if**(**isset**($\_SESSION[**'login'**]))  
 **echo '<form>  
 <div class="form-group">  
 <textarea class="form-control" rows="3"></textarea>  
 </div>  
 <button class="btn btn-dark" id="sendComment">Залишити відгук</button>  
 </form>'**;  
 **?>** </**div**>  
  
  
 <**a class="btn btn-dark" id="getMoreComments" style="color**:**white"**>Ще коментарі</**a**>  
 </**div**>  
 </**div**>  
  
 *<!--  
  
  
 -->  
 <!-- /.card -->*

</**div**>  
 *<!-- /.col-lg-9 -->* </**div**>  
</**div**>  
**<?php  
if**(**isset**($\_SESSION[**'login'**]))  
 **echo '<script src="/templates/modules/rating/rating.js"></script>'**;  
**?>**<**script src="/templates/modules/comments/commentsController.js"**></**script**>  
  
*<!-- /.container -->*

*<!-- Page Content -->***<?php** *extract*($bookInfo)**?>**<**div class="container"**>  
 <**div class="row"**>  
 *<!--  
  
 <h1>***<?=**$id**?>***</h1>  
 <h1>***<?=**$name**?>***</h1>  
 -->* <**input type="hidden" id="bookid" value="<?=**$id**?>"**>  
 <**div class="col-lg-2"**>  
  
 </**div**>  
 *<!-- /.col-lg-3 -->* <**div class="col-lg-8"**>  
  
 <**div class="card mt-4"**>  
 <**img class="mainimg adminform" src="/uploads/books/<?=**$imageId**?>" alt=""**>  
  
 <**div class="card-body"**>  
 <**h3 class="card-title"**>**<?=**$name**?>**</**h3**>  
 <**div**>  
 <**h4**>**<?=**$price**?>** грн.</**h4**>  
 <**button type="button" value="<?=**$id**?>" class="btn btn-default btn-md addtocart"** >  
 <**span class="fas fa-shopping-cart"**></**span**> Додати в кошик  
 </**button**>  
 </**div**>  
 <**br**>  
 <**table class="table"**>  
 <**tr**>  
 <**td**>Рік видання</**td**>  
 <**td**>**<?=**$year**?>**</**td**>  
 </**tr**>  
 <**tr**>  
 <**td**>Кількість сторінок</**td**>  
 <**td**>**<?=**$pages**?>**</**td**>  
 </**tr**>  
 <**tr**>  
 <**td**>Мова</**td**>  
 <**td**>**<?=**$language**?>**</**td**>  
 </**tr**>  
 </**table**>  
 <**p class="card-text"**>**<?=**$description**?>**</**p**>  
  
 **<?php  
 require "templates/modules/rating/stars.tpl.php"**;  
 **?>** </**div**>  
 </**div**>  
 *<!-- /.card -->* <**div class="card card-outline-secondary my-4"**>  
 <**div class="card-header"**>  
 Product Reviews  
 </**div**>  
  
 <**div class="card-body" id="comment-container"**>  
 <**div**>  
 **<?php  
 if**(**isset**($\_SESSION[**'login'**]))  
 **echo '<form>  
 <div class="form-group">  
 <textarea class="form-control" rows="3"></textarea>  
 </div>  
 <button class="btn btn-dark" id="sendComment">Залишити відгук</button>  
 </form>'**;  
 **?>** </**div**>  
  
  
 <**a class="btn btn-dark" id="getMoreComments" style="color**:**white"**>Ще коментарі</**a**>  
 </**div**>  
 </**div**></**div**>  
 *<!-- /.col-lg-9 -->* </**div**>  
  
</**div**>  
**<?php  
if**(**isset**($\_SESSION[**'login'**]))  
 **echo '<script src="/templates/modules/rating/rating.js"></script>'**;  
**?>**<**script src="/templates/modules/comments/commentsController.js"**></**script**>  
  
*<!-- /.container -->*

Лістинг «кошику»

<**div class="container" style="margin-top**:100**px"**>  
 <**form action="/orders/order" method="post"**>  
 **<?php** $i = 0;  
 **foreach** ($orderList **as** $key => $item) { *// hz naschet* **if**($i==$key){  
 **echo '<div class="col-lg-5 form-signin cartst" data-id="'**.$item[**'id'**].**'">'**;  
 **echo "<p class='form-control'>"**.$item[**'name'**].**"</p>"**;  
 **echo "<p class='form-control'>"**.$item[**'author'**].**"</p>"**;  
 **echo "<p class='price'>"**.$item[**'price'**].**"</p>"**;

**echo "<input data-id='"**.$item[**'id'**].**"' class='form-control' name='"**.$item[**'id'**].**"amount' value='"**.$item[**'amount'**].**"'>"**;

**echo '</div>'**;  
 }  
 $i++;  
 }  
  
  
 **echo '<div class = "">'**;  
 **echo '<div class="form-signint cartst col-lg-12">'**;  
 **echo '<p class="total-cost" style="text-align: center">Загальна вартість: '**.$totalCost.**' грн.</p>'**;  
 **echo '<input type="address" value="м.Житомир вул. Вокзальна 34, кв.21" class="full-width">'**;  
 **echo '<br>'**;  
 **echo '<input type="submit" class="btn btn-default btn-block" value="Оформити замовлення">'**;  
 **echo '</div>'**;  
 **echo '</div>'**;  
  
 **?>** </**form**>  
</**div**>  
</**div**>  
<**script**>  
 $(**"form input"**).focusout(**function** () {  
 **var** cost = 0;  
 $(**"form input[data-id]"**).each(**function** () {  
 **var** price = $(**'p.price'**,$(**this**).parent()).text();  
  
 **var** amount = +$(**this**).val();  
 cost += price\*amount;  
 });  
 $(**"p.total-cost"**).text(cost+**' грн.'**)  
 });  
  
 $(**"form"**).submit(**function** (ev) {  
 ***console***.log(1);  
 ev.preventDefault();  
 **var** arr = [];  
 $(**"div input"**,**this**).each(**function** () {  
 **for**(**var** i=0;i<$(**this**).val();i++)  
 arr.push($(**this**).attr(**'data-id'**));  
 });  
 *setCookie*(**'order'**,***JSON***.stringify(arr),1000);  
  
 $.ajax({  
 **url**: **'/orders/order'**,  
 **data**: {  
 },  
 **type**: **'post'**,  
 success: **function** (res) {  
  
 ***console***.log(res);  
 *deleteCookie*(**'order'**);  
 ***document***.location.href=**"/users/profile"**;  
 }  
 });  
 });  
  
 **function** *setCookie*(name, value, options) {  
 options = options || {};  
  
 **var** expires = options.**expires**;  
  
 **if** (**typeof** expires == **"number"** && expires) {  
 **var** d = **new *Date***();  
 d.setTime(d.getTime() + expires \* 1000);  
 expires = options.**expires** = d;  
 }  
 **if** (expires && expires.toUTCString) {  
 options.**expires** = expires.toUTCString();  
 }  
  
 value = *encodeURIComponent*(value);  
  
 **var** updatedCookie = name + **"="** + value+**";path=/"**;  
  
 **for** (**var** propName **in** options) {  
 updatedCookie += **"; "** + propName;  
 **var** propValue = options[propName];  
 **if** (propValue !== **true**) {  
 updatedCookie += **"="** + propValue;  
 }  
 }  
  
 ***document***.cookie = updatedCookie;  
 }  
  
 **function** *deleteCookie*(name) {  
 *setCookie* (name, **""**, {  
 **expires**: -1  
 })  
 }  
</**script**>

Лістинг форми реєстрації

<**div class="bglib"**>  
<**div class = "container"**>  
 <**div class="wrapper slimform"**>  
 <**form method="post" name="Registry\_Form" class="form-signin"**>  
 <**h3 class="form-signin-heading"**>Реєстрація</**h3**>  
 <**hr class="colorgraph"**><**br**>  
  
 <**input type="text" class="form-control" name="Username" placeholder="Логін" autofocus=""** />  
 <**input type="password" class="form-control" name="Password" placeholder="Пароль"** />  
 <**input type="password" class="form-control" name="PasswordCheck" placeholder="Повторіть пароль"** />  
 <**input type="date" class="form-control" name="Birthday" placeholder="День народження"** />  
 <**input type="email" class="form-control" name="Email" placeholder="E-mail"** />  
 <**input type="button" class="btn btn-default btn-block btcolor" id="submitbutton" value="Зареєструватись"**>  
 </**form**>  
 </**div**>  
</**div**>  
</**div**>  
<**script src="/templates/modules/users/registrationFieldsChecker.js"**></**script**>

Лістинг форми авторизації

<**div class="bglib"**>  
<**div class = "container"**>  
  
 <**div class="wrapper slimform"**>  
 **<?php  
 if**($registered==1)  
 **echo '<span style="text-align: center">Підтверддіть реєстрацію на Вашій пошті</span>'**;  
 **?>** <**form action="/users/login" method="post" name="Login\_Form" class="form-signin"**>  
 <**h3 class="form-signin-heading"**>Раді бачити Вас знову! Будь-ласка, авторизуйтесь:</**h3**>  
 <**hr class="colorgraph"**><**br**>  
  
 <**input type="text" class="form-control" name="Username" placeholder="Логін" required="" autofocus=""** />  
 <**input type="password" class="form-control" name="Password" placeholder="Пароль" required=""**/>  
 <**input type="button" class="btn btn-default btn-block btcolor" value="Увійти" id="loginButton"**>  
  
 </**form**>  
 </**div**>  
</**div**>  
</**div**>  
<**script src="/templates/modules/users/loginChecker.js"**></**script**>

Лістинг сторінки профілю користувача

<**div class="container"**>  
  
 <**h2**>Раді знову бачити тебе, **<?=**$\_SESSION[**'login'**]**?>**</**h2**>  
   
 <**ul class="historycash"**>  
  
 <**table class="table"**>  
 <**thead**>  
 <**tr**>  
 <**th scope="col"**>#</**th**>  
 <**th scope="col"**>Автор</**th**>  
 <**th scope="col"**>Назва</**th**>  
 <**th scope="col"**>Ціна</**th**>  
 <**th scope="col"**>Кількість</**th**>  
 </**tr**>  
 </**thead**>  
 <**tbody**>  
 **<?php  
 foreach** ($orders **as** $item) {  
 **echo '<tr><td colspan="5" style="text-align: center;">Номер замовлення - '**.$item[**'orderid'**].**'</td></tr>'**;  
 $i=0;  
  
 **foreach** ($item[**'ordered'**] **as** $key=>$t) {  
 $t = (array)$t;  
  
 **echo '<tr>'**;  
 **echo '<th scope="row">'** . $i . **'</th>  
   
 <td>'**.$t[**'author'**].**'</td>  
 <td>'**.$t[**'name'**].**'</td>  
 <td>'**.$t[**'price'**].**'</td>  
 <td>'** .$t[**'amount'**]. **'</td>'**;  
 **echo '</tr>'**;  
 $i++;  
 **if**($key==*count*($item[**'ordered'**])-2)  
 **break**;  
 }  
 }  
 **?>** </**tbody**>  
 </**table**>  
</**div**>

Лістинг форми додавання товару

<**div class="container"**>  
  
 <**div class="row adminform"**>  
  
 <**div class=" col-lg-6"**>  
 <**h3**>Додати нову книгу:</**h3**>  
 <**form action="/catalog/addBook" method="post" enctype="multipart/form-data"**>  
 <**input type="hidden" name="action" value="newbook"**>  
 <**ul class="list-group add-form"**>  
 <**li class="list-group-item"**>  
 <**p class="key"**>Автор:</**p**>  
 <**p class="val"**>  
 <**input type="text" name="author" class="full-width"**>  
 </**p**>  
 </**li**>  
  
 <**li class="list-group-item"**>  
 <**p class="key"**>Назва:</**p**>  
 <**p class="val"**>  
 <**input type="text" name="name" class="full-width"**>  
 </**p**>  
 </**li**>  
  
 <**li class="list-group-item"**>  
 <**p class="key"**>Ціна:</**p**>  
 <**p class="val"**>  
 <**input type="text" name="price" class="full-width"**>  
 </**p**>  
 </**li**>  
  
 <**li class="list-group-item"**>  
 <**p class="key"**>Жанри: (розділіть ';' )</**p**>  
 <**p class="val"**>  
 <**input type="text" name="genres" class="full-width"**>  
 </**p**>  
 </**li**>  
  
 <**li class="list-group-item"**>  
 <**p class="key"**>Кількість:</**p**>  
 <**p class="val"**>  
 <**input type="number" name="amount" class="full-width"**>  
 </**p**>

</**li**>  
  
  
 <**li class="list-group-item"**>  
 <**p class="key"**>Опис:</**p**>  
 <**p class="val"**>  
 <**textarea name="description" cols="30" rows="10" class="form-control"**></**textarea**>  
 </**p**>  
 </**li**>  
  
  
 <**li class="list-group-item"**>  
 <**p class="key"**>Рік:</**p**>  
 <**p class="val"**>  
 <**input type="number" name="year" class="full-width"**>  
 </**p**>  
 </**li**>  
  
 <**li class="list-group-item"**>  
 <**p class="key"**>Вікове обмеження:</**p**>  
 <**p class="val"**>  
 <**input type="number" name="age" class="full-width"**>  
 </**p**>  
 </**li**>  
  
 <**li class="list-group-item"**>  
 <**p class="key"**>Кількість сторінок:</**p**>  
 <**p class="val"**>  
 <**input type="number" name="pages" class="full-width"**>  
 </**p**>  
 </**li**>  
  
 <**li class="list-group-item"**>  
 <**p class="key"**>Мова:</**p**>  
 <**p class="val"**>  
 <**input type="text" name="language" class="full-width"**>  
 </**p**>  
 </**li**>  
  
 <**li class="list-group-item"**>  
 <**p class="key"**>Фото:</**p**>  
 <**p class="val"**>  
 <**input type="file" name="img"** >  
 </**p**>  
 </**li**>  
  
 </**ul**>  
  
  
 <**input type="submit" class="btn btn-default" value="Додати книгу"**>  
 </**form**>  
  
 *<!-- amount form -->* </**div**>  
 <**div class="col-lg-6"**>  
 <**h3**>Додати товар на складі:</**h3**>  
 <**ul class="list-group add-form"**>  
 <**form action="/catalog/addBook" method="post"**>  
 <**input type="hidden" name="action" value="amount"**>

<**li class="list-group-item"**>  
 <**p class="key"**>Пошук:</**p**>  
 <**p class="val"**>  
 <**input type="text" name="searchfield" class="full-width"**>  
 </**p**>  
 </**li**>  
  
 <**li class="list-group-item"**>  
 <**p class="key"**>Книга:</**p**>  
 <**p class="val"**>  
 <**select name="bookslist" class="full-width"**>  
 **<?php  
 foreach** ($books **as** $book)  
 **echo '<option value="'**.$book[**'id'**].**'">'**.$book[**'author'**].**' - '**.$book[**'name'**].**'</option>'  
 ?>** </**select**>  
 </**p**>  
 </**li**>  
  
 <**li class="list-group-item"**>  
 <**p class="key"**>Кількість:</**p**>  
 <**p class="val"**>  
 <**input type="number" name="amount" class="full-width"**>  
 </**p**>  
 </**li**>  
  
 <**input type="submit" value="Додати товар" class="btn btn-default"**>  
 </**form**>  
 </**ul**>  
 </**div**>  
 </**div**>  
 <**br**>  
</**div**>  
  
<**script**>  
 $(**"input[name='searchfield']"**).focusout(**function** () {  
 **var** searchString = $(**this**).val();  
 ***console***.log(searchString);  
 $(**"select[name='bookslist'] option"**).each(**function** () {  
 $(**this**).removeAttr(**'selected'**);  
 **if**($(**this**).text().toLowerCase().indexOf(searchString.toLowerCase()) >= 0)  
 {  
 $(**this**).attr(**'selected'**,**true**);  
 ***id*** = $(**this**).val();  
 }  
 });  
  
  
 });  
</**script**>

Лістинг форми зміни/видалення товару

<**div class="container"**>  
 <**div class="row adminform"**>  
 <**div class=" col-lg-6 erclass"**>  
 <**ul**>  
 <**h2**>Скористейтесь пошуком для знаходження необхідної книги</**h2**>  
 <**form action="/catalog/updateBook" method="post" class="edit" enctype="multipart/form-data"**>  
 <**li class="list-group-item"**>  
 <**p class="key"**>Пошук:</**p**>  
 <**p class="val"**>  
 <**input type="text" name="searchfield" class="full-width"**>  
 </**p**>  
 </**li**>  
  
 <**li class="list-group-item"**>  
 <**p class="key"**>Автор:</**p**>  
 <**p class="val"**>  
 <**select name="bookslist" class="full-width"**>  
 **<?php  
 foreach** ($books **as** $book)  
 **echo '<option value="'**.$book[**'id'**].**'">'**.$book[**'id'**].**' - '**.$book[**'author'**].**' - '**.$book[**'name'**].**'</option>'  
 ?>** </**select**>  
 </**li**>  
 <**input type="hidden" name="id" value="<?=**$book[**'id'**]**?>"**>  
 <**li class="list-group-item"**>  
 <**p class="key"**>Автор:</**p**>  
 <**p class="val"**>  
 <**input type="text" name="author" class="full-width"**>  
 </**p**>  
 </**li**>  
 <**li class="list-group-item"**>  
 <**p class="key"**>Назва:</**p**>  
 <**p class="val"**>  
 <**input type="text" name="name" class="full-width"**>  
 </**p**>  
 </**li**>  
  
 <**li class="list-group-item"**>  
 <**p class="key"**>Ціна:</**p**>  
 <**p class="val"**>  
 <**input type="text" name="price" class="full-width"**>  
 </**p**>  
 </**li**>  
  
 <**li class="list-group-item"**>  
 <**p class="key"**>Жанри:</**p**>  
 <**p class="val"**>  
 <**input type="text" name="genres" class="full-width"**>  
 </**p**>  
 </**li**>  
  
 <**li class="list-group-item"**>  
 <**p class="key"**>Кількість:</**p**>  
 <**p class="val"**>  
 <**input type="number" name="amount" class="full-width"**>  
 </**p**>  
 </**li**>  
  
 <**li class="list-group-item"**>  
 <**p class="key"**>Опис:</**p**>  
 <**p class="val"**>  
 <**textarea name="description" cols="30" rows="10" class="form-control"**></**textarea**>  
 </**p**>  
 </**li**>  
  
  
 <**li class="list-group-item"**>  
 <**p class="key"**>Рік видання:</**p**>  
 <**p class="val"**>  
 <**input type="number" name="year" class="full-width"**>  
 </**p**>  
 </**li**>  
  
 <**li class="list-group-item"**>  
 <**p class="key"**>Вікове обмеження:</**p**>  
 <**p class="val"**>  
 <**input type="number" name="age" class="full-width"**>  
 </**p**>  
 </**li**>  
  
 <**li class="list-group-item"**>  
 <**p class="key"**>Кількість сторінок:</**p**>  
 <**p class="val"**>  
 <**input type="number" name="pages" class="full-width"**>  
  
 </**p**>  
 </**li**>  
  
 <**li class="list-group-item"**>  
 <**p class="key"**>Мова:</**p**>  
 <**p class="val"**>  
 <**input type="text" name="language" class="full-width"**><**br**>  
 </**p**>  
 </**li**>  
  
 <**li class="list-group-item"**>  
 <**p class="key"**>Фото:</**p**>  
 <**p class="val"**>  
 <**img class="mainimg" src="" alt=""**>  
 </**p**>  
 </**li**>  
  
 <**br**>  
  
 <**input type="file" name="img"**>  
 <**br**>  
 <**br**>  
 <**input type="submit" value="Оновити" class="btn btn-default"** >  
  
 </**form**>  
 <**br**>  
 <**form action="/catalog/deleteBook" method="post" class="delete"**>  
 <**input type="hidden" name="id" value="<?=**$book[**'id'**]**?>"**>  
 <**input type="submit" value="Видалити книгу" class="btn btn-default"** >  
 </**form**>  
 </**ul**>  
 <**hr**>

**<?php  
 foreach** ($books **as** $book)  
 **echo '<option value="'**.$book[**'id'**].**'">'**.$book[**'author'**].**' - '**.$book[**'name'**].**'</option>'  
 ?>**</**div**>  
 </**div**>  
</**div**>  
<**script**>  
 $(**"form.delete"**).submit(**function** (ev) {  
 ev.preventDefault();  
 **id** = $(**'input[name="id"]'**).val();  
 $(**"option[value="**+**id**+**"]"**).remove();  
 $.ajax({  
 **url**: **'/catalog/deleteBook'**,  
 **data**: {  
 **id**:**id** },  
 **type**: **'post'**,  
 success: **function** (res) {  
  
 *loadSelectedBook*();  
 }  
 });  
 **id**=$(**"select[name='bookslist'] option:first-child"**).val();  
  
 });  
  
</**script**>  
<**script src="/templates/modules/catalog/bookSearch.js"**></**script**>

Додаток Г

Файл DB.inc.php

**<?php  
class** DB  
{  
 **protected $pdo**;  
 **public function** \_\_construct($host,$user,$pass,$dbname)  
 {  
 $this->**pdo** = **new** PDO(**"mysql:host=**{$host}**;dbname=**{$dbname}**"**,$user,$pass);  
 }  
  
 **public function** RegisterUser($row)  
 {  
 $stmt = $this->**pdo**->prepare(**'insert into users(login,email,password,token,birthdate) values(:login,:email,:password,:token,:birthdate)'**);  
 $stmt->bindParam(**':login'**,$row[**'login'**]);  
 $stmt->bindParam(**':password'**,$row[**'password'**]);  
 $stmt->bindParam(**':birthdate'**,$row[**'bday'**]);  
 $stmt->bindParam(**':token'**,$row[**'token'**]);  
 $stmt->bindParam(**':email'**,$row[**'email'**]);  
 $stmt->execute();  
 }  
  
 **public function** LoginUser($login,$password)  
 {  
 $stmt = $this->**pdo**->prepare(**'select** *count***(***\****) from users where login=:l and password=:p and status=1'**);  
 $stmt->bindParam(**':l'**,$login);  
 $stmt->bindParam(**':p'**,$password);  
 $stmt->execute();  
  
 $res = $stmt->fetch();  
 **return** $res[0];  
 }  
  
 **public function** CheckEmail($email)  
 {  
 $stmt = $this->**pdo**->prepare(**'select** *count***(***\****) from users where email=:email'**);  
 $stmt->bindParam(**':email'**,$email);  
 $stmt->execute();  
  
 $res = $stmt->fetch();  
 **return** $res[0];  
 }  
  
 **public function** CheckLogin($login)  
 {  
 $stmt = $this->**pdo**->prepare(**'select** *count***(***\****) from users where login=:login'**);  
 $stmt->bindParam(**':login'**,$login);  
 $stmt->execute();  
  
 $res = $stmt->fetch();  
 **return** $res[0];  
 }  
  
 **public function** ActivateAccount($token)  
 {

$stmt = $this->**pdo**->prepare(**'update users set status=1 where token=:token '**);  
 $stmt->bindParam(**':token'**,$token);  
 $stmt->execute();  
 }  
  
 **public function** GetUserRole($login)  
 {  
 $stmt = $this->**pdo**->prepare(**'select role from users where login=:login'**);  
 $stmt->bindParam(**':login'**,$login);  
 $stmt->execute();  
  
 $res = $stmt->fetch();  
 **return** $res[0];  
 }  
  
 **public function** AddBook($row)  
 {  
 $stmt = $this->**pdo**->prepare(**'insert into books (author,name,price,genre,description,year, age, pages, language,imageId) values(:author,:name,:price,:genre,:description,:year, :age, :pages, :language, :imageId)'**);  
 $stmt->bindParam(**':name'**,$row[**'name'**]);  
 $stmt->bindParam(**':author'**,$row[**'author'**]);  
 $stmt->bindParam(**':price'**,$row[**'price'**]);  
 $stmt->bindParam(**':genre'**,$row[**'genres'**]);  
 $stmt->bindParam(**':description'**,$row[**'description'**]);  
 $stmt->bindParam(**':year'**,$row[**'year'**]);  
 $stmt->bindParam(**':age'**,$row[**'age'**]);  
 $stmt->bindParam(**':pages'**,$row[**'pages'**]);  
 $stmt->bindParam(**':language'**,$row[**'language'**]);  
 $stmt->bindParam(**':imageId'**,$row[**'imageId'**]);  
 $stmt->execute();  
  
  
 $stmt = $this->**pdo**->prepare(**'SELECT LAST\_INSERT\_ID() as id'**);  
 $stmt->execute();  
 $bookid = $stmt->fetchAll()[0][0];  
  
  
 $stmt = $this->**pdo**->prepare(**'insert into storage (bookid,amount) values (:bookid,:amount)'**);  
 $stmt->bindParam(**':bookid'**,$bookid);  
 $stmt->bindParam(**':amount'**,$row[**'amount'**]);  
 $stmt->execute();  
  
  
 }  
  
 **public function** IncreaseAmount($row)  
 {  
 $stmt = $this->**pdo**->prepare(**'update storage set amount=amount+:amount where bookid=:bookid'**);  
 $stmt->bindParam(**':bookid'**,$row[**'bookid'**]);  
 $stmt->bindParam(**':amount'**,$row[**'amount'**]);  
 $stmt->execute();  
 }  
  
 **public function** GetAllBooks($offset,$amount,$genre)  
 {  
 **if**(*gettype*($genre)==**'integer'**)  
 {  
 $stmt = $this->**pdo**->prepare(**'select id,name,author,price,imageid from books limit :limit offset :offset '**);  
 }  
 **else**{  
 $stmt = $this->**pdo**->prepare(**'select id,name,author,price,imageid from books where genre like :genre limit :limit offset :offset '**);  
 $temp = **'%'**.$genre.**'%'**;  
  
  
 $stmt->bindParam(**':genre'**,$genre);  
 }  
  
 $amount = *intval*($amount);  
 $offset = *intval*($offset);  
  
 $stmt->bindParam(**':limit'**,$amount,PDO::***PARAM\_INT***);  
 $stmt->bindParam(**':offset'**,$offset,PDO::***PARAM\_INT***);  
 $stmt->execute();  
 $res = $stmt->fetchAll();  
  
  
  
  
 **return** $res;  
 }  
  
 **public function** GetAllGenres()  
 {  
 $stmt = $this->**pdo**->prepare(**'SELECT GROUP\_CONCAT(genre SEPARATOR \'; \')FROM books '**);  
 $stmt->execute();  
 $res = $stmt->fetchAll();  
 **return** $res;  
 }  
  
 **public function** UpdateBook($row)  
 {  
 $stmt = $this->**pdo**->prepare(**'update books set author=:author,name=:name,price=:price,  
genre=:genre,description=:description, year=:year, age=:age, pages=:pages, language=:language, imageid=:imageid where id=:id'**);  
 $stmt->bindParam(**':id'**,$row[**'id'**]);  
 $stmt->bindParam(**':author'**,$row[**'author'**]);  
 $stmt->bindParam(**':name'**,$row[**'name'**]);  
 $stmt->bindParam(**':price'**,$row[**'price'**]);  
 $stmt->bindParam(**':genre'**,$row[**'genres'**]);  
 $stmt->bindParam(**':description'**,$row[**'description'**]);  
 $stmt->bindParam(**':year'**,$row[**'year'**]);  
 $stmt->bindParam(**':age'**,$row[**'age'**]);  
 $stmt->bindParam(**':pages'**,$row[**'pages'**]);  
 $stmt->bindParam(**':language'**,$row[**'language'**]);  
 $stmt->bindParam(**':imageid'**,$row[**'imageId'**]);  
  
 $stmt->execute();  
 }  
  
 **public function** DeleteBook($id)  
 {  
 $stmt = $this->**pdo**->prepare(**'delete from books where id=:id'**);  
 $stmt->bindParam(**':id'**,$id);  
 $stmt->execute();  
 $res = $stmt->fetchAll();  
 **return** $res;  
 }  
  
 **public function** GetAmount($id)  
 {  
 $stmt = $this->**pdo**->prepare(**'SELECT amount from storage where bookid=:id '**);  
 $stmt->bindParam(**':id'**,$id);  
 $stmt->execute();  
 $res = $stmt->fetchAll()[0];  
 **return** $res;  
 }  
  
 **public function** GetBookById($id)  
 {  
 $stmt = $this->**pdo**->prepare(**'SELECT** *\** **from books where id=:bid '**);  
 $stmt->bindParam(**':bid'**,$id);  
 $stmt->execute();  
 $res = $stmt->fetchAll();  
 **return** $res;  
 }  
  
 **public function** GetRating($bookid,$userlogin)  
 {  
 $stmt = $this->**pdo**->prepare(**'select** *avg***(mark) as avgMark from marks where bookid=:bookid'**);  
 $stmt->bindParam(**':bookid'**,$bookid);  
 $stmt->execute();  
 $res = [];  
 $res[**'avgMark'**] = *round*( $stmt->fetchAll()[0][0]);  
  
 **if**($userlogin!=**null** && $userlogin!=**''**){  
 $stmt = $this->**pdo**->prepare(**'select id from users where login=:login'**);  
 $stmt->bindParam(**':login'**,$userlogin);  
 $stmt->execute();  
 $userid = $stmt->fetchAll()[0][**'id'**];  
  
 **if**($userid!=**null** && $userid!=0)  
 {  
 $stmt = $this->**pdo**->prepare(**'select mark from marks where bookid=:bookid and userid=:userid'**);  
 $stmt->bindParam(**':userid'**,$userid);  
 $stmt->bindParam(**':bookid'**,$bookid);  
 $stmt->execute();  
 $m = $stmt->fetchAll();  
 **if**(*count*($m)>0)  
 $res[**'userRating'**] = *intval*($m[0][0]);  
 **else** $res[**'userRating'**] = 0;  
 }  
 }  
 **return** $res;  
 }  
  
 **public function** setRating($bookid, $userlogin, $rating)  
 {  
 $stmt = $this->**pdo**->prepare(**'select id from users where login=:login'**);  
 $stmt->bindParam(**':login'**, $userlogin);  
 $stmt->execute();  
 $userid = $stmt->fetchAll()[0][**'id'**];  
  
 $stmt = $this->**pdo**->prepare(**'select** *count***(***\****) from marks where bookid=:bookid and userid=:userid'**);  
 $stmt->bindParam(**':bookid'**, $bookid);  
 $stmt->bindParam(**':userid'**, $userid);  
 $stmt->execute();  
 $res = $stmt->fetchAll();  
  
 **if** ($res[0][0] == 0){  
  
 $stmt = $this->**pdo**->prepare(**'insert into marks(bookid,userid,mark) values(:bookid,:userid,:mark)'**);  
 $stmt->bindParam(**':bookid'**,$bookid);  
 $stmt->bindParam(**':userid'**,$userid);  
 $stmt->bindParam(**':mark'**,$rating);  
 $stmt->execute();  
 }  
 **else**{  
  
 $stmt = $this->**pdo**->prepare(**'update marks set mark=:mark where bookid=:bookid and userid=:userid'**);  
 $stmt->bindParam(**':bookid'**,$bookid);  
 $stmt->bindParam(**':userid'**,$userid);  
 $stmt->bindParam(**':mark'**,$rating);  
 $stmt->execute();  
  
 }  
  
 **return** $stmt->errorInfo();  
 }  
  
 **public function** GetUserIdByLogin($login)  
 {  
 $stmt = $this->**pdo**->prepare(**'select id from users where login=:login'**);  
 $stmt->bindParam(**':login'**, $login);  
 $stmt->execute();  
 $userid = $stmt->fetchAll();  
 **return** $userid;  
 }  
  
 **public function** ConfirmOrder($row)  
 {  
 $stmt = $this->**pdo**->prepare(**'insert into orders(userid,ordered,price,date) values(:userid,:ordered,:price,:date)'**);  
 $stmt->bindParam(**':userid'**,$row[**'userid'**]);  
 $stmt->bindParam(**':ordered'**,$row[**'ordered'**]);  
 $stmt->bindParam(**':price'**,$row[**'price'**]);  
 $stmt->bindParam(**':date'**,$row[**'date'**]);  
 $stmt->execute();  
 *var\_dump*($stmt->errorInfo());  
 }  
  
 **public function** GetAllUserOrders($id)  
 {  
 $stmt = $this->**pdo**->prepare(**'select** *\** **from orders where userid=:userid'**);  
 $stmt->bindParam(**':userid'**, $id);  
 $stmt->execute();  
 $res = $stmt->fetchAll();  
 **return** $res ;  
 }  
  
 **public function** AddComment($row)  
 {  
 $userid = $this->GetUserIdByLogin($row[**'userLogin'**])[0][0];  
  
  
 $stmt = $this->**pdo**->prepare(**'insert into comments(userid,bookid,content,datetime) values(:userid,:bookid,:content,:datetime)'**);  
 $stmt->bindParam(**':userid'**,$userid);  
 $stmt->bindParam(**':bookid'**,$row[**'id'**]);

$stmt->bindParam(**':content'**,$row[**'content'**]);  
 $stmt->bindParam(**':datetime'**,$row[**'datetime'**]);  
 $stmt->execute();  
 $res = $stmt->fetchAll();  
 **return** $res;  
 }  
  
 **public function** GetComments($row)  
 {$stmt = $this->**pdo**->prepare(**'select comments.id,comments.userid,comments.content,DATE\_FORMAT(comments.datetime, \'%d.%m.%Y %H:%i:%s\') as datetime,users.login from comments   
 inner join books on books.id = comments.bookid  
 inner join users on users.id = comments.userid  
 where bookid=:bookid order by DATE\_FORMAT(comments.datetime, \'%d.%m.%Y %H:%i:%s\') desc limit :limit offset :offset '**);  
 $id = *intval*($row[**'id'**]);  
 $limit = *intval*($row[**'commentsN'**]);  
 $offset = *intval*($row[**'offset'**]);  
 $stmt->bindParam(**':bookid'**,$id);  
 $stmt->bindParam(**':limit'**,$limit,PDO::***PARAM\_INT***);  
 $stmt->bindParam(**':offset'**,$offset,PDO::***PARAM\_INT***);  
 $stmt->execute();  
 $res = $stmt->fetchAll();  
  
 **return** $res;  
 }  
  
}

Додаток Д

Файл Template.inc.php

**<?php  
class** Template  
{  
 **protected $Params**;  
 **protected $FileName**;  
 **protected $ContentView**;  
 **protected $isAjaxRequest** = **false**;  
  
 **public function** \_\_construct($fileName)  
 {  
 $this->**Params** = **array**();  
 $this->**FileName** = $fileName;  
 }  
  
 **public function** AjaxRequest()  
 {  
 $this->**isAjaxRequest**=**true**;  
 }  
  
 **public function** setParam($name,$value)  
 {  
 **if**(!**empty**($name))  
 $this->**Params**[$name] = $value;  
 }  
  
 **public function** setContentView($contentView)  
 {  
 **if**(**empty**($contentView))  
 $this->**ContentView** = **'templates/index.tpl.php'**;  
 **else** $this->**ContentView** =$contentView;  
  
 $this->setParam(**'content\_view'**,$this->**ContentView**);  
 }  
  
 **public function** setParams($arr)  
 {  
  
 **if**(!**empty**($arr))  
 $this->**Params** = *array\_merge*($this->**Params**,$arr);  
 }  
 **public function** fetch()  
 {  
 *extract*($this->**Params**);  
 *ob\_start*();  
 **include**($this->**FileName**);  
 $html = *ob\_get\_contents*();  
 *ob\_end\_clean*();  
 **return** $html;  
 }  
  
 **public function** display()  
 {  
 **if**(!$this->**isAjaxRequest**)  
 **echo** $this->fetch();  
 }  
}  
**?>**

Додаток Е

**Каталог**

Controller

**<?php  
class** Catalog\_Controller  
{  
 **public function** ListAction()  
 {  
 $genres = Catalog\_Model::*GetAllGenres*();  
  
 Core::*$IndexTpl*->setParam(**'genres'**,$genres);  
 Catalog\_View::*listView*();  
 }  
  
 **public function** AddBookAction()  
 {  
 $hasRights = **true**;  
 **if**(!**isset**($\_SESSION[**'role'**]))  
 {  
 $hasRights=**false**;  
 }  
 **else  
 if**($\_SESSION[**'role'**]!=**'admin'**)  
 $hasRights=**false**;  
  
 **if**(!$hasRights)  
 {  
 $mc = **new** Main\_Controller();  
 $mc->Error404Action();  
 **return**;  
 }  
  
  
 **switch**($\_SERVER[**'REQUEST\_METHOD'**])  
 {  
 **case 'GET'**:  
 Catalog\_View::*AddBookView*();  
 $books = Catalog\_Model::*GetAllBooks*(0,100000,0);  
 Core::*$IndexTpl*->setParam(**'books'**,$books);  
 **break**;  
 **case 'POST'**:  
  
 $books = Catalog\_Model::*GetAllBooks*(0,100000,0);  
 Core::*$IndexTpl*->setParam(**'books'**,$books);  
 **switch** ($\_POST[**'action'**])  
 {  
  
 **case 'newbook'**:  
 Catalog\_Model::*AddBook*($\_POST,$\_FILES);  
  
  
 **break**;  
 **case 'amount'**:  
  
 *//Catalog\_Model::GetAllBooks();* Catalog\_Model::*IncreaseAmount*($\_POST);  
 **break**;  
  
 }  
  
 Catalog\_View::*AddBookView*();  
 **break**;  
 }  
 }  
  
 **public function** EditBookAction()  
 {  
 $hasRights = **true**;  
 **if**(!**isset**($\_SESSION[**'role'**]))  
 {  
 $hasRights=**false**;  
 }  
 **else  
 if**($\_SESSION[**'role'**]!=**'admin'**)  
 $hasRights=**false**;  
  
 **if**(!$hasRights)  
 {  
 $mc = **new** Main\_Controller();  
 $mc->Error404Action();  
 **return**;  
 }  
  
 Catalog\_View::*EditBookView*();  
 $books = Catalog\_Model::*GetAllBooks*(0,100000,0);  
 Core::*$IndexTpl*->setParam(**'books'**,$books);  
 }  
  
 **public function** UpdateBookAction()  
 {  
 Catalog\_Model::*UpdateBook*($\_POST,$\_FILES);  
  
 *header*(**'Location: /catalog/editBook'**);  
 }  
  
 **public function** GetBooksAction()  
 {  
 Core::*SetIsAjaxRequest*();  
  
 **if**(!**isset**($\_POST[**'genre'**]))  
 $books = Catalog\_Model::*GetAllBooks*($\_POST[**'offset'**],$\_POST[**'amount'**],0);  
 **else** {  
 $\_POST[**'genre'**] = **"%"**.*ucfirst*($\_POST[**'genre'**]).**"%"**;  
  
 $books = Catalog\_Model::*GetAllBooks*($\_POST[**'offset'**],$\_POST[**'amount'**],$\_POST[**'genre'**]);  
 }  
 $res = *json\_encode*($books);  
  
 **echo** $res;  
 }  
  
 **public function** ViewAction($params)  
 {  
 $id = *array\_shift*($params);  
  
 $bookInfo = Catalog\_Model::*GetBookById*($id);  
  
 **if**($bookInfo[**'amount'**]==**null**){  
 $mc = **new** Main\_Controller();

$mc->Error404Action();  
 **return**;  
 }  
  
 **if**(**isset**($\_SESSION[**'login'**]))  
 $temp = Rating\_Model::*GetRating*($bookInfo[**'id'**],$\_SESSION[**'login'**]);  
 **else** $temp = Rating\_Model::*GetRating*($bookInfo[**'id'**],**''**);  
  
 **if**(**isset**($\_SESSION[**'login'**]))  
 $bookInfo[**'userMark'**]= $temp[**'userRating'**];  
  
 $bookInfo[**'avgMark'**] = $temp[**'avgMark'**];  
 $bookInfo[**'userMark'**]= $temp[**'userRating'**];  
  
 Catalog\_View::*BookInfoView*();  
 Core::*$IndexTpl*->setParam(**'bookInfo'**,$bookInfo);  
  
 }  
  
 **public function** GetBookAction()  
 {  
 Core::*SetIsAjaxRequest*();  
 $hasRights = **true**;  
 **if**(!**isset**($\_SESSION[**'role'**]))  
 {  
 $hasRights=**false**;  
 }  
 **else  
 if**($\_SESSION[**'role'**]!=**'admin'**)  
 $hasRights=**false**;  
  
 **if**(!$hasRights)  
 {  
 $mc = **new** Main\_Controller();  
 $mc->Error404Action();  
 **return**;  
 }  
  
  
 $res = Catalog\_Model::*GetBookById*($\_POST[**'id'**]);  
 **echo** *json\_encode*($res);  
 }  
  
 **public function** DeleteBookAction()  
 {  
 Core::*SetIsAjaxRequest*();  
 Catalog\_Model::*DeleteBook*($\_POST[**'id'**]);  
 }  
}

Model

**<?php  
class** Catalog\_Model{  
  
 **public static function** AddBook($row,$files)  
 {  
 **if**(**isset**($files[**'img'**]) && $files[**'img'**][**'size'**]!=0) {  
 $poster = $files[**'img'**];  
 $row[**'imageId'**] = *uniqid*() . **'.'** . *explode*(**'/'**, $poster[**'type'**])[1];  
 $target = **'uploads/books/'** . $row[**'imageId'**];  
 *move\_uploaded\_file*($poster[**'tmp\_name'**], $target);  
 }

$db = Core::*GetDB*();  
  
 $res = $db->AddBook($row);  
 }  
  
 **public static function** IncreaseAmount($row)  
 {  
 $db = Core::*GetDB*();  
  
 $res = $db->IncreaseAmount($row);  
  
 }  
  
 **public static function** GetAllBooks($offset,$amount,$genre)  
 {  
 $db = Core::*GetDB*();  
  
 $res = $db->GetAllBooks($offset,$amount,$genre);  
 **foreach** ($res **as** $key => $book)  
 {  
 **if**(**isset**($\_SESSION[**'login'**]))  
 $ulogin = $\_SESSION[**'login'**];  
 **else** $ulogin=**''**;  
 $temp = Rating\_Model::*GetRating*($book[**'id'**],$ulogin);  
 $res[$key][**'avgMark'**] = $temp[**'avgMark'**];  
 **if**(**isset**($\_SESSION[**'login'**]))  
 $res[$key][**'userMark'**]= $temp[**'userRating'**];  
 $res[$key][**'imageid'**]=$book[**'imageid'**];  
 }  
  
  
 **return** $res;  
 }  
  
 **public static function** UpdateBook($row,$files)  
 {  
 **if**(**isset**($files[**'img'**]) && $files[**'img'**][**'size'**]!=0)  
 {  
 $poster = $files[**'img'**];  
 $row[**'imageId'**] = *uniqid*().**'.'**.*explode*(**'/'**,$poster[**'type'**])[1];  
 $target = **'uploads/books/'**.$row[**'imageId'**];  
 *move\_uploaded\_file*($poster[**'tmp\_name'**],$target);  
 }  
 $db = Core::*GetDB*();  
  
 $res = $db->UpdateBook($row);  
 }  
  
 **public static function** DeleteBook($id)  
 {  
 $db = Core::*GetDB*();  
  
 $res = $db->DeleteBook($id);  
  
 }  
  
 **public static function** GetAllGenres()  
 {  
 $db = Core::*GetDB*();

$res = $db->GetAllGenres();  
  
  
 $res = *array\_map*(**'ucfirst'**, *explode*(**'; '**,$res[0][0]));  
 $res = *array\_unique*($res);  
  
 **return** $res;  
  
 }  
  
 **public static function** GetBookById($id)  
 {  
 $db = Core::*GetDB*();  
  
 $res = $db->GetBookById($id)[0];  
  
 $res[**'amount'**] = $db->GetAmount($id)[0];  
  
 **return** $res;  
 }  
}

View

**<?php  
class** Catalog\_View  
{  
 **public static function** listView()  
 {  
 Core::*SetMainContent*(**'templates/modules/catalog/catalog.tpl.php'**);  
 }  
  
 **public static function** AddBookView()  
 {  
 Core::*SetMainContent*(**'templates/modules/catalog/addBook.tpl.php'**);  
 }  
  
 **public static function** BookInfoView()  
 {  
  
 Core::*SetMainContent*(**'templates/modules/items/item.tpl.php'**);  
 }  
 *//EditBookView* **public static function** EditBookView()  
 {  
 Core::*SetMainContent*(**'templates/modules/catalog/editBook.tpl.php'**);  
 }  
 *//*}

Додаток Є

Коментарі

Controller

**<?php  
class** Comments\_Controller  
{  
 **public function** AddCommentAction()  
 {  
 Core::*SetIsAjaxRequest*();  
 $res = Comments\_Model::*addComment*($\_POST);  
 **echo** *json\_encode*($res);;  
 }  
  
 **public function** GetCommentsToFilmAction()  
 {  
 Core::*SetIsAjaxRequest*();  
 $comments = Comments\_Model::*getComments*($\_POST);  
 $res = *json\_encode*($comments);  
 **echo** $res;  
 }  
  
}

Model

**<?php  
class** Comments\_Model  
{  
 **public static function** AddComment($row)  
 {  
 $db = Core::*GetDB*();  
  
 $datetime = *explode*(**' '**,$row[**'datetime'**]);  
  
 $date = *explode*(**'.'**,$datetime[0]);  
 $row[**'datetime'**] = $date[2].**'-'**.$date[1].**'-'**.$date[0].**' '**.$datetime[1];*//+* $db->AddComment($row);  
  
 }  
  
 **public static function** getComments($row)  
 {  
 $db = Core::*GetDB*();  
  
 $res = $db->GetComments($row);  
  
  
 **return** $res;  
 }  
}

Додаток Ж

Головна сторінка

Controller

**<?php  
class** Main\_Controller  
{  
 **public function** IndexAction()  
 {  
 Main\_View::*indexView*();  
 }  
  
 **public function** Error404Action()  
 {  
 Core::*$IndexTpl*->setContentView(**'templates/error404.tpl'**);  
 }  
  
}

Model

**<?php  
class** Main\_Model  
{  
 **public static function** getCurrentMoviesPreview()  
 {  
 $db = **new** DB(***DB\_HOST***,***DB\_USER***,***DB\_PASS***,***DB\_NAME***);  
 $res = $db->GetCurrentMoviesPreview();  
  
 **foreach** ($res **as** $key => $item){  
 $temp = $db->GetSessionsByMovieId($item[**'id'**]);  
  
 **foreach** ($temp **as** $kkey => $val) {  
 **if**(!**isset**($res[$key][**'sessions'**][$val[**'date'**]]))  
 $res[$key][**'sessions'**][$val[**'date'**]] = **array**();  
 }  
  
 **foreach** ($temp **as** $kkey => $val){  
 *array\_push*($res[$key][**'sessions'**][$val[**'date'**]],**array**(**"time"**=> $val[**'time'**],**'id'**=>$val[**'id'**]));  
  
 }  
 *//$res[$key]['sessions'] = $db->GetSessionsByMovieId($item['id']);* }  
 **return** $res;  
 }  
}

View

**<?php  
class** Main\_View  
{  
 **public static function** indexView()  
 {  
 $tpl = Core::*$IndexTpl*;  
 $tpl->setContentView(**'templates/index.tpl.php'**);  
 }  
}

Додаток З

Замовлення

Controller

**<?php  
class** Orders\_Controller  
{  
 **public function** CartAction($params)  
 {  
 **if**(!**isset**($\_COOKIE[**'order'**]))  
 {  
 $m = **new** Main\_Controller();  
 $m->Error404Action();  
 **return**;  
 }  
 $arr = *json\_decode*($\_COOKIE[**'order'**]);  
 $arr = *array\_count\_values*($arr );  
 $res = **array**();  
 $i = 0;  
 $totalCost = 0;  
 **foreach** ($arr **as** $key => $value){  
 $res[$i] = Catalog\_Model::*GetBookById*($key);  
 $res[$i][**'amount'**] = $value;  
 $totalCost += $res[$i][**'amount'**]\**intval*($res[$i][**'price'**]);  
 $i++;  
 }  
 $res[**'totalCost'**] = $totalCost;  
  
 Core::*SetMainContent*(**'templates/modules/orders/cart.tpl.php'**);  
 Core::*$IndexTpl*->setParam(**'orderList'**,$res);  
 Core::*$IndexTpl*->setParam(**'totalCost'**,$totalCost);  
 }  
  
 **public function** OrderAction()  
 {  
 $arr = *json\_decode*($\_COOKIE[**'order'**]);  
 $arr = *array\_count\_values*($arr );  
 $res = **array**();  
 $i = 0;  
 $totalCost = 0;  
 **foreach** ($arr **as** $key => $value){  
 $res[$i] = Catalog\_Model::*GetBookById*($key);  
 $res[$i][**'amount'**] = $value;  
 $totalCost += $res[$i][**'amount'**]\**intval*($res[$i][**'price'**]);  
 $i++;  
 }  
 $res[**'price'**] = $totalCost;  
  
 Orders\_Model::*ConfirmOrder*($res);  
  
 }  
  
}

Model

**<?php  
class** Orders\_Model  
{  
 **public static function** ConfirmOrder($res)  
 {  
 $db = Core::*GetDB*();  
  
 $save[**'ordered'**] = *json\_encode*($res);  
 $save[**'date'**] = *date*(**"Y-m-d"**);  
 $save[**'userid'**] = $db->GetUserIdByLogin($\_SESSION[**'login'**])[0][0];  
 $save[**'price'**] = $res[**'price'**];  
 $db->ConfirmOrder($save);  
  
 }  
  
 **public static function** GetAllUserOrders($res)  
 {  
 $db = Core::*GetDB*();  
  
 $userid = $db->GetUserIdByLogin($\_SESSION[**'login'**])[0][0];  
  
 $res = $db->GetAllUserOrders($userid);  
 $arr = [];  
 $i = 0;  
 **foreach** ($res **as** $item)  
 {  
 $arr[$i] = [];  
 $arr[$i][**'ordered'**] = (array)*json\_decode*($item[**'ordered'**]);  
 $arr[$i][**'date'**] = $item[**'date'**];  
 $arr[$i][**'orderid'**] = $item[**'id'**];  
  
 $i++;  
 }  
  
 **return** $arr;  
 }  
}

Додаток И

Оцінювання

Controller

**<?php  
class** Rating\_Controller  
{  
 **public function** SetRatingAction()  
 {  
 Core::*SetIsAjaxRequest*();  
  
  
 $res = Rating\_Model::*setRating*($\_POST[**'id'**],$\_POST[**'userLogin'**],$\_POST[**'mark'**]);  
  
  
 **echo** *json\_encode*($res);  
 }  
  
 **public function** GetRatingAction()  
 {  
 Core::*SetIsAjaxRequest*();  
  
  
 $comments = Rating\_Model::*getRating*($\_POST[**'id'**],$\_POST[**'userLogin'**]);  
  
 $res = *json\_encode*($comments);  
  
 **echo** $res;  
 }  
  
}

Model

**<?php  
class** Rating\_Model  
{  
 **public static function** GetRating($id,$userlogin)  
 {  
 $db = Core::*GetDB*();  
  
 $res = $db->GetRating($id,$userlogin);  
  
 **return** $res;  
 }  
  
 **public static function** SetRating($id,$userlogin,$mark)  
 {  
 $db = Core::*GetDB*();  
  
 $res = $db->SetRating($id,$userlogin,$mark);  
  
 **return** $res;  
 }  
}

Додаток I

Користувачі

Controller

**<?php  
class** Users\_Controller{  
  
 **public function** LoginAction($params)  
 {  
 **switch**($\_SERVER[**'REQUEST\_METHOD'**])  
 {  
 **case 'GET'**:  
 Users\_View::*loginView*();  
 $temp = *array\_shift*($params);  
 **if**($temp==1)  
 Core::*$IndexTpl*->setParam(**'registered'**,1);  
 **else** Core::*$IndexTpl*->setParam(**'registered'**,0);  
 **break**;  
 **case 'POST'**:  
 Core::*SetIsAjaxRequest*();  
 $res = Users\_Model::*LoginUser*($\_POST[**'login'**],$\_POST[**'password'**]);  
  
 **if**($res==1)  
 **echo** 1;  
 **else  
 echo** -1;  
 **break**;  
 }  
 }  
  
 **public function** RegistrationAction()  
 {  
 **switch**($\_SERVER[**'REQUEST\_METHOD'**])  
 {  
 **case 'GET'**:  
 Users\_View::*registerView*();  
 **break**;  
 **case 'POST'**:  
 Core::*SetIsAjaxRequest*();  
 $res = Users\_Model::*CheckInputs*($\_POST);  
  
 **echo** *json\_encode*($res);  
 **if**(*count*($res)==0)  
 Users\_Model::*registerUser*($\_POST);  
 **break**;  
 }  
 }  
  
 **public function** ActivateAction($params)  
 {  
 $token = *array\_shift*($params);  
 Users\_Model::*ActivateAccount*($token);  
 }  
  
 **public function** LogoutAction()  
 {  
 **if**(**isset**($\_COOKIE[**'order'**]))  
 {  
 **unset**($\_COOKIE[**'order'**]);  
 *setcookie*(**'order'**, **null**, -1, **'/'**);  
 }  
 *session\_unset*();  
 *session\_destroy*();  
 *header*(**'Location: /users/login/ '**);  
 }  
  
 **public function** ProfileAction()  
 {  
 $orders = Orders\_Model::*GetAllUserOrders*($\_SESSION[**'login'**]);  
  
 Core::*$IndexTpl*->setParam(**'orders'**,$orders);  
 Users\_View::*ProfileView*();  
  
 *// var\_dump($orders);die();* }  
}

Model

**<?php  
class** Users\_Model{  
  
 **public static function** ActivateAccount($token)  
 {  
 $db = Core::*GetDB*();  
 $db->ActivateAccount($token);  
 *header*(**"Location: /users/login/0"**);  
 }  
  
 **public static function** SendMail($email,$theme,$message)  
 {  
 $headers = **"Content-Type: text/html; charset=ISO-8859-1\r\n"**;  
  
 *mail*($email,$theme,$message,$headers);  
 }  
  
 **public static function** registerUser($row)  
 {  
 $db = Core::*GetDB*();  
  
 $row[**'token'**] = *md5*(*uniqid*());  
 $row[**'password'**] = *md5*($row[**'password'**]);  
  
 $db->RegisterUser($row);  
  
 $message = **'<a href="http://'** . $\_SERVER[**"HTTP\_HOST"**] . **'/users/activate/'** . $row[**'token'**] . **'">Click here to activate your account<a/> '**;  
 $message .= **"<br/><br/>If you recieved this message by accident just ignore it."**;  
 $message = *wordwrap*($message, 70, **"\r\n"**);  
  
  
 **self**::*sendMail*($row[**'email'**],**'Account activation'**, $message);  
 }  
  
 **public static function** LoginUser($login, $password)  
 {  
 $db = Core::*GetDB*();  
 $res = $db->LoginUser($login,*md5*($password));

**if**($res == 1){  
 $\_SESSION[**'login'**]=$login;  
 $\_SESSION[**'role'**]=$db->GetUserRole($login);  
  
 *// $\_SESSION['role']=$res['role'];* **return** 1;  
 }  
 **else  
 return** -1;  
 }  
  
  
 **public static function** CheckInputs($row)  
 {  
 */\*data: {  
 login: $("input[name='Username']").val(),  
 password: $("input[name='Password']").val(),  
 bday: $("input[name='Birthday']").val(),  
 email: $("input[name='Email']").val()  
 },\*/* $db = Core::*GetDB*();  
 $returnObj = [];  
  
 $res = $db->CheckEmail($row[**'email'**]);  
 **if**($res==1)  
 $returnObj[**'email'**]=-1;  
  
 $res = $db->CheckLogin($row[**'login'**]);  
 **if**($res==1)  
 $returnObj[**'login'**]=-1;  
  
 **return** $returnObj;  
  
 }  
  
}

View

**<?php  
class** Users\_View  
{  
 **public static function** loginView()  
 {  
 Core::*SetMainContent*(**'templates/modules/users/login.tpl.php'**);  
  
 }  
  
 **public static function** registerView()  
 {  
 Core::*SetMainContent*(**'templates/modules/users/registration.tpl.php'**);  
  
  
 }  
  
 **public static function** ProfileView()  
 {  
 Core::*SetMainContent*(**'templates/modules/users/profile.tpl.php'**);  
 }  
}

Додаток Ї

Файли JS.

bookLoader.js

**var *genres*** = [];  
**var *amount*** = 3;  
**var *offset*** = 0;  
**var *isloading*** = **false**;  
**var *selectedGenre*** =**''**;  
  
***window***.onload = **function** () {  
  
 $(**"a.genre"**).each(**function** () {  
 ***genres***.push($(**this**).text());  
 });  
 ***selectedGenre*** = $(**"a.genre:first-child"**).attr(**'data-genre'**);  
 ***console***.log(***selectedGenre***);  
 *loadBooks*();  
};  
  
  
**function** *loadBooks*(genreChanged) {  
  
 ***isloading*** = **true**;  
 $.ajax({ **url**: **'/catalog/getBooks'**,  
 **data**: {  
 **offset**: ***offset***,  
 **amount**: ***amount***,  
 **genre**: ***selectedGenre*** },  
 **type**: **'post'**,  
 success: **function**(output) {  
  
 **if**(genreChanged!=**undefined** && genreChanged==**true**)  
 $(**"#items div.row"**).empty();  
  
 **var** arr = ***JSON***.parse(output);  
 ***console***.log(arr);  
 ***offset***+=arr.**length**;  
 **if**(arr.**length**==0)  
 $(**"#loadmore"**).attr(**'disabled'**,**true**);  
  
  
 $(arr).each(**function** () {  
 ***console***.log(**this**);  
 **var** rating = $(**"<div id='"**+**this**[**'id'**]+**"'></div>"**);  
 **for**(**var** i=0;i<5;i++){  
 **if**(i<**this**[**'avgMark'**])  
 rating.append($(**"<span class=\"fa fa-star checked\"></span>"**));  
 **else** rating.append($(**"<span class=\"fa fa-star fa\"></span>"**));  
 }  
  
 **var** $item = $(**'<a href="/catalog/view/'**+**this**[**'id'**]+**'"><div class="col-lg-4 col-md-6 mb-4">\n'** +  
 **' <div class="card h-100">\n'**+  
 **' <img class="card-img-top" style="" src="/uploads/books/'**+**this**[**'imageid'**]+**'" alt="">\n'** +  
 **' <div class="card-body">\n'** +  
 **' <h4 class="card-title">\n'** +  
 **' <a href="/catalog/view/'**+**this**[**'id'**]+**'">'**+**this**[**'name'**]+**'</a>\n'** +  
 **' </h4>\n'** +  
 **' <h5>'**+**this**[**'author'**]+**'</h5>\n'** +  
 **' <h5>'**+**this**[**'price'**]+**' грн</h5>\n'** +  
 **' </div>\n'** +  
 **' <div class="card-footer">\n'** +  
  
 **' </div>\n'** +  
 **' </div>\n'** +  
 **' </div></a>'**);  
 $(**"div.card-footer"**,$item).append(rating);  
 $(**"#items div.row"**).append($item);  
  
 });  
 }  
 });  
}  
  
$(**"#loadmore"**).click(*loadBooks*);  
  
  
$(**"a.genre"**).click(**function** (ev) {  
 ***offset***=0;  
 $(**"a.genre"**).removeClass(**'selected'**);  
 $(**this**).addClass(**'selected'**);  
 ***selectedGenre*** = $(**"a.genre.selected"**).attr(**'data-genre'**);  
 ***console***.log(***selectedGenre***);  
  
 *loadBooks*(**true**);  
  
 $(**"#loadmore"**).attr(**'disabled'**,**false**);  
});

bookSearch.js

**var *id***;  
  
$(**"select"**).change(**function** () {  
 ***id***=**this**.**value**;  
 $(**"input[name='id']"**).val(***id***);  
 ***console***.log(***id***);  
 *loadSelectedBook*();  
});  
  
***window***.onload = **function** () {  
 ***id***=$(**"select[name='bookslist'] option:first-child"**).val();  
  
 $(**"input[name='id']"**).val(***id*** );  
 *loadSelectedBook*();  
};  
  
$(**"input[name='searchfield']"**).focusout(**function** () {  
 **var** searchString = $(**this**).val();  
 ***console***.log(searchString);  
 $(**"select[name='bookslist'] option"**).each(**function** () {  
 $(**this**).removeAttr(**'selected'**);  
 **if**($(**this**).text().toLowerCase().indexOf(searchString.toLowerCase()) >= 0)  
 {  
 $(**this**).attr(**'selected'**,**true**);  
 ***id*** = $(**this**).val();  
 }  
 });  
  
 *loadSelectedBook*();  
});  
  
**function** *loadSelectedBook*() {  
 $.ajax({  
 **url**: **'/catalog/getBook'**,  
 **data**: {  
 **id**:***id*** },  
 **type**: **'post'**,  
 success: **function** (res) {  
 **var** book = ***JSON***.parse(res);  
 ***console***.log(book);  
  
 $(**"input[name='genres']"**).val(book[**'genre'**]);  
 ***console***.log($(**"input[name='id']"**).val());  
 $(**"input[name='id']"**).val(book[**'id'**]);  
  
  
 $(**"input[name='author']"**).val(book[**'author'**]);  
 **var** t = **'http://'**+***document***.location.host+**'/uploads/books/'**+book[**'imageId'**];  
 $(**'img'**).attr(**'src'**,t);  
 ***console***.log(t);  
 $(**"input[name='age']"**).val(book[**'age'**]);  
 $(**"textarea[name='description']"**).val(book[**'description'**]);  
 $(**"input[name='language']"**).val(book[**'language'**]);  
 $(**"input[name='name']"**).val(book[**'name'**]);  
 $(**"input[name='pages']"**).val(book[**'pages'**]);  
 $(**"input[name='year']"**).val(book[**'year'**]);  
 $(**"input[name='price']"**).val(book[**'price'**]);  
 $(**"input[name='amount']"**).val(book[**'amount'**]);  
 }  
 })  
}

orderController.js

$(**"button.addtocart"**).click(**function** () {  
 *//deleteCookie('order');* **if**(*getCookie*(**'order'**)==**undefined**)  
 **var** arr = [];  
 **else  
 var** arr = ***JSON***.parse(*getCookie*(**'order'**));  
 ***console***.log(arr);  
 arr.push($(**"#bookid"**).val());  
 *setCookie*(**'order'**,***JSON***.stringify(arr),100);  
 **if**(arr.length==1)  
 $(**"ul.navbar-nav.ml-auto"**).append($(**"<li class=\"nav-item\">\n"** +  
 **" <a class=\"nav-link js-scroll-trigger\" href='/orders/cart'>Cart ( <span>1</span> )</a>\n"** +  
 **" </li>"**));  
 **else**{  
 ***console***.log();  
 **var** t = +$($(**"ul.navbar-nav.ml-auto:last-child a span"**)[2]).text()+1;  
  
 $($(**"ul.navbar-nav.ml-auto:last-child a span"**)[2]).text(t);  
 }  
});  
  
**function** *setCookie*(name, value, options) {  
 options = options || {};  
  
 **var** expires = options.**expires**;  
  
 **if** (**typeof** expires == **"number"** && expires) {  
 **var** d = **new *Date***();  
 d.setTime(d.getTime() + expires \* 1000);  
 expires = options.**expires** = d;  
 }  
 **if** (expires && expires.toUTCString) {  
 options.**expires** = expires.toUTCString();  
 }  
  
 value = *encodeURIComponent*(value);  
  
 **var** updatedCookie = name + **"="** + value+**";path=/"**;  
  
 **for** (**var** propName **in** options) {  
 updatedCookie += **"; "** + propName;  
 **var** propValue = options[propName];  
 **if** (propValue !== **true**) {  
 updatedCookie += **"="** + propValue;  
 }  
 }  
  
 ***document***.cookie = updatedCookie;  
}  
  
  
**function** *getCookie*(name) {  
 **var** matches = ***document***.cookie.match(**new *RegExp***(  
 **"(?:^|; )"** + name.replace(/([\.$?\*|{}\(\)\[\]\\\/\+^])/g, **'\\$1'**) + **"=([^;]\*)"** ));  
 **return** matches ? *decodeURIComponent*(matches[1]) : **undefined**;  
}  
  
**function** *deleteCookie*(name) {  
 *setCookie* (name, **""**, {  
 **expires**: -1  
 })  
}  
  
**function** *onlyUnique*(value, index, self) {  
 **return** self.indexOf(value) === index;  
}

commentsControlle.js

**var *offset*** = 0;  
**var *commentsN*** = 5;  
**var *bookid***;  
***window***.onload = **function** () {  
 ***bookid*** = $(**"#bookid"**).val();  
 $(**"#getMoreComments"**).click(*getComments*);  
 *getComments*();  
};  
  
$(**"#sendComment"**).click(**function** (ev) {  
 ev.preventDefault();  
 **var** userLogin = $(**"#userlogin"**).text();  
 **var** d = **new *Date***();  
 **var** options = {  
 **year**: **'numeric'**,  
 **month**: **'numeric'**,  
 **day**: **'numeric'**,  
 **hour**: **'numeric'**,  
 **minute**: **'numeric'**,  
 **second**: **'numeric'** };  
 **var** datetime = d.toLocaleString(**'uk'**, options).replace(**','**, **''**);  
 **var** content = $(**"textarea"**).val();  
 **if** (content == **''**)  
 **return**;  
  
  
  
 **var** $comment = $(**"<div>\n"** +  
  
 **" <p>"**+content+**"</p>\n"** +  
 **" <small class=\"text-muted\">Posted by "**+userLogin+**" on "**+datetime+**"</small>\n"** +  
 **" <hr>\n"** +  
 **" </div>"**);  
  
 $(**"#comment-container div.form-group"**).parent().parent().append($comment);  
  
  
 $comment.fadeIn(**'slow'**);  
 $.ajax({  
 **url**: **'/comments/addComment'**,  
 **data**: {  
 **id**: ***bookid***,  
 **userLogin**: userLogin,  
 **datetime**: datetime,  
 **content**: content  
 },  
 **type**: **'POST'**,  
 success: **function** (res) {  
 ***console***.log( res);  
 *//console.log( JSON.parse(res));* }  
 });  
});  
**function** *getComments*() {  
 $.ajax({ **url**: **'/comments/getCommentsToFilm'**,  
 **data**: {  
 **id**: ***bookid***,  
 **offset**: ***offset***,  
 **commentsN**: ***commentsN*** },  
 **type**: **'POST'**,  
 success: **function**(res) {  
 **var** arr = ***JSON***.parse(res);  
 ***console***.log(arr);  
 ***offset***+= ***commentsN***;  
 **for**(**var** i=0;i<arr.**length**;i++){  
 **var** $comment = $(**"<div>\n"** +  
 **" <p>"**+arr[i][**'content'**]+**"</p>\n"** +  
 **" <small class=\"text-muted\">Posted by "**+arr[i][**'login'**]+**" on "**+arr[i][**'datetime'**]+**"</small>\n"** +  
 **" <hr>\n"** +  
 **" </div>"**);  
 $comment.insertBefore($(**"#getMoreComments"**));

$comment.fadeIn(**'slow'**);  
 }  
  
 }  
 });  
}

Rating.js

$(**"span.fa.fa-star"**).hover(**function** () {  
  
 **var** indexOflast = $(**this**).index();  
 $(**"span.fa.fa-star"**,$(**this**).parent()).each(**function** () {  
 $(**this**).removeClass(**'checked'**);  
 **if**($(**this**).index()<=indexOflast)  
 {  
 $(**this**).addClass(**'checked'**);  
 }  
 });  
});  
  
$(**"span.fa.fa-star"**).parent().mouseout(**function** () {  
  
 $(**"span.fa.fa-star"**,$(**this**).parent()).each(**function** () {  
 **if**($(**this**).index()<=$(**".bookmark"**,$(**this**).parent()).val())  
 {  
 $(**this**).addClass(**'checked'**);  
 }  
 **else**{  
 $(**this**).removeClass(**'checked'**);  
 }  
  
 });  
});  
**var *context***;  
  
$(**"span.fa.fa-star"**).click(**function** () {  
 **var** id = $(**this**).parent().attr(**'id'**);  
 ***context*** = **this**;  
 **var** mark = $(**this**).index();  
 $.ajax({  
 **url**: **'/rating/setRating'**,  
 **data**: {  
 **id**: id,  
 **userLogin**: $(**"#userlogin"**).text(),  
 **mark**: mark  
 },  
 **type**: **'POST'**,  
 success: **function** (res) {  
 $.ajax({  
 **url**: **'/rating/getRating'**,  
 **data**: {  
 **id**: id,  
 **userLogin**: $(**"#userlogin"**).text()  
 },  
 **type**: **'POST'**,  
 success: **function** (res) {  
  
 **var** newavg = ***Math***.round(***JSON***.parse( res)[**'avgMark'**]);  
 $(**".bookmark"**,$(***context***).parent()).val(newavg);  
  
 $(**"span:nth-last-child(2)"**,$(***context***).parent()).text(**'Середня оцінка: '**+***Math***.round(***JSON***.parse( res)[**'avgMark'**]));  
 $(**"span:last-child"**,$(***context***).parent()).text(**'Ваша оцінка: '**+***JSON***.parse( res)[**'userRating'**]);  
 $(**"span.fa.fa-star"**,$(***context***).parent()).each(**function** () {  
 $(**this**).removeClass(**'checked'**);  
 **if**($(**this**).index()<=newavg)  
 {  
 $(**this**).addClass(**'checked'**);  
 }  
 });  
  
 }  
 });  
  
 }  
 });  
});

loginCheker.jss’   
$(**"#loginButton"**).click(**function** () {  
 $.ajax({ **url**: **'/users/login'**,  
 **data**: {  
 **login**: $(**"input[name='Username']"**).val(),  
 **password**: $(**"input[name='Password']"**).val()  
 },  
 **type**: **'post'**,  
 success: **function**(output) {  
 ***console***.log(output);  
 **if**(output==-1) {  
 *alert*(**'wrong login or password'**);  
 }  
 **else**{  
 *///redirect to login* ***document***.location.href=**"/"**;  
 }  
 }  
 });  
});

RegistrationfieldsChecker.js

$(**"#submitbutton"**).click(**function** () {  
 **if**($(**"input[name='Password']"**).val()!=$(**"input[name='PasswordCheck']"**).val())  
 {  
 *alert*(**'confirm password is incorrect'**);  
 **return**;  
 }  
 $(**this**).prop(**'disabled'**, **true**);  
 $.ajax({ **url**: **'/users/registration'**,  
 **data**: {  
 **login**: $(**"input[name='Username']"**).val(),  
 **password**: $(**"input[name='Password']"**).val(),  
 **bday**: $(**"input[name='Birthday']"**).val(),  
 **email**: $(**"input[name='Email']"**).val()  
 },  
 **type**: **'post'**,  
 success: **function**(output) {  
 $(**"#submitbutton"**).prop(**'disabled'**, **false**);  
 **var** res = ***JSON***.parse(output);  
 **if**(res.**length**!=0) {  
 **if** (res.hasOwnProperty(**'login'**)) {  
 **if** (res[**'login'**] == -1)  
 *alert*(**'login already used'**);  
 }  
 **if** (res.hasOwnProperty(**'email'**)) {  
 **if** (res[**'email'**] == -1)  
 *alert*(**'email already used'**);  
 }  
  
 }  
 **else**{  
 *///redirect to login* ***document***.location.href=**"/users/login/1"**;  
 }  
 }  
 });  
});

Додаток Й

Таблиця стилів

**header** {  
 **padding**: 154**px** 0 100**px**;  
}  
  
**@media** (**min-width**: 992**px**) {  
 **header** {  
 **padding**: 156**px** 0 100**px**;  
 }  
}  
  
**section** {  
 **padding**: 150**px** 0;  
}  
  
.**erclass**{  
 **margin-left**: 25%;  
}  
  
**body**{  
 **margin-top**: -25**px**;  
 **font-family**: **Courier New**, **monospace**,**bold**;  
}  
  
.**wrapper** {  
 **margin-top**: 80**px**;  
 **margin-bottom**: 20**px**;  
}  
  
.**form-signin**,.**form-signint** {  
 **width**:88%;  
 **padding**: 30**px** 38**px** 66**px**;  
 **margin**: 0 **auto**;  
 **background-color**: **#eee**;  
 **border**: 3**px dotted rgba**(0,0,0,0.1);  
}  
  
.**form-signin-heading** {  
 **text-align**:**center**;  
 **margin-bottom**: 30**px**;  
}  
  
.**form-control** {  
 **position**: **relative**;  
 **font-size**: 16**px**;  
 **height**: **auto**;  
 **padding**: 10**px**;  
}  
  
**input**[**type**=**"text"**] {  
 **margin-bottom**: 0**px**;  
 */\* border-bottom-left-radius: 0;  
 border-bottom-right-radius: 0;\*/*}  
  
**input**[**type**=**"password"**] {  
 **margin-bottom**: 20**px**;  
 */\* border-top-left-radius: 0;  
 border-top-right-radius: 0;\*/*}  
  
.**colorgraph** {  
 **height**: 2**px**;  
 **border-top**: 0;  
 **background**: **maroon**;  
 **border-radius**: 5**px**;  
}  
  
.**book-preview-maket**{  
 **display**: **none**;  
}  
  
**a**.**selected** {  
 **color**: **red**;  
}  
  
**span**.**checked** {  
 **color**: **orange**;  
}  
  
**span**.**marks** {  
 **display**:**block**;  
}  
  
**input**.**full-width**,**select**.**full-width**{  
 **width**:100%;  
}  
  
.**nav-item a**:**hover**{  
 **text-decoration**: **none**;  
}  
  
.**bglib**{  
 **height**: 100%;  
 **background-image**: **url**(**"/uploads/bg/lib.jpg"**);  
}  
  
.**margtop**{  
 **margin-top**: 0;  
}  
.**btcolor**{  
 **background-color**: **brown**;  
}  
**a**, **a**:**hover**,**a**:**visited**{  
 **color**: **maroon**;  
}  
  
.**cartst**{  
 **border**: 2**px solid lightcyan**;  
 **margin-top**: 50**px**;  
 **margin-left**: 5%;  
 **display**: **inline-block**;  
}  
  
**ul**.**historycash** {  
 **list-style**: **square outside**}  
  
**button**{  
 **margin-bottom**: 10**px**;  
 **margin-top**: 5**px**;  
}  
.**mainimg**{  
 **width**: 40%;  
 **height**: 50%;  
 **margin-left**: 30%;  
}  
.**adminform**{  
 **margin-top**: 100**px**;  
}  
  
.**slimform**{  
 **width**: 500**px**;  
 **height**: 600**px**;  
 **margin-left**: 27%;  
}  
  
.**smalltable**{  
 **margin**: 5**px**;  
 **padding**: 2**px**;  
 **border**: 1**px solid lightgray**;  
}