



ИНТЕГРИРАНА УЕБ-БАЗИРАНА ПЛАТФОРМА  
ЗА ОБУЧЕНИЕ НА СТУДЕНТИ  
ПО БИБЛИОТЕЧНО-ИНФОРМАЦИОННИ НАУКИ:  
АРХИТЕКТУРА И ИМПЛЕМЕНТАЦИЯ

Павел Митков Горанов

AN INTEGRATED WEB-BASED PLATFORM  
FOR STUDENT LIBRARY  
AND INFORMATION SCIENCES EDUCATION:  
ARCHITECTURE AND IMPLEMENTATION

Pavel Mitkov Goranov

**Abstract:** This paper presents an integrated web-based learning platform for library and information science students, focusing on its architecture and implementation. The platform provides capabilities for creating bibliographic records, bibliographic references, and lists, based on the MARC 21 standard for bibliographic description. Students can view bibliographic records in ISBD format, ensuring completeness and accuracy of information resources. The developed system provides an intuitive user interface and complies with modern digital learning requirements, facilitating the professional development of future librarians.

**Keywords:** keyword list in English, web-based platform, tuition, MARC 21, ISBD, library and information sciences, digital learning

## ВЪВЕДЕНИЕ

С развитието на информационните технологии и дигитализацията на библиотечната дейност, обучението на студентите по библиотечно-информационни науки изисква съвременни методи и инструменти за ефективно предаване на знания и умения. Традиционните образователни подходи постепенно отстъпват място на уеб-базираните платформи, които предлагат по-голяма гъвкавост, достъпност и интерактивност. В този контекст, необходимостта от иновативни решения, които да съчетават стандартите за библиографско описание с възможностите на дигиталната среда, става все по-належаща.

Въвеждането на такива платформи е ключово за подготовката на бъдещите библиотекари в условията на бързо развиващата се информационна ера. Настоящата работа цели не само да опише техническите аспекти на разработената система, но и да покаже нейното значение за образователния процес и бъдещите възможности за развитие и подобрене на платформата.

Целта на този доклад е да представи архитектурата и имплементацията на интегрирана уеб-базирана платформа за обучение на студенти по библиотечно-информационни науки. Платформата е създадена с оглед осигуряването на функционалности за създаване на библиографски

записи, библиографски справки и библиографски списъци, базирани на стандарта MARC 21 за библиографско описание. Също така, тя предлага възможност за преглед на библиографските записи в ISBD формат, като осигурява пълнота и коректност при представянето на информацията.

## **Изложение**

### **1. Теоретични основи на библиографското описание**

Библиографското описание представлява съвкупност от стандартизирани процедури и правила, които се използват за създаване на библиографски записи, съдържащи детайлна информация за различни видове документи и ресурси. Тези записи имат ключово значение за управлението на библиотечните колекции, като улесняват търсенето, идентифицирането и достъпа до информационните ресурси. Библиографското описание е важен елемент от библиотечното дело и информацията, която съдържа, е структурирана по определени стандарти, за да може да бъде разбираема и еднозначна както за хората, така и за компютърните системи.

Създаването на правилна и коректна каталогизация се основава на международните принципи на каталогизацията, известни под името „Парижки принципи“, които съществуват още от 1961 г. и които ИФЛА преработва през 2009 г. с цел адаптирането им към нуждите на общодостъпните каталози в режим онлайн през 21 в. Една от основните цели на принципите е постигането на унифициран подход към библиографското описание. Общите принципи са удобство на ползвателя на каталога; общоприета терминология – използваните при описанията и достъпа термини трябва да отговарят на представата на мнозинството от ползвателите; представяне – описанията и контролираните форми на имената (наименованията) трябва да се основават на начина, по който обекта описва себе си; точност – описваният обект трябва да бъде отразен достоверно; достатъчност и необходимост. В описанията и контролираните форми на имената (наименованията), предназначени за достъп, трябва да се включват само онези елементи, които са необходими за подпомагане на ползвателя и са от значение за идентифицирането на обекта; значимост – елементите трябва да бъдат значими от библиографска гледна точка; икономичност – когато целта може да бъде постигната по различни начини, избира се най-икономичния подход (напр., най-евтиния или най-простия); последователност и стандартизация – описанията и оформянето на точки за достъп трябва да се стандартизират, доколкото е възможно. Така се осигурява по-голяма последователност, която на свой ред обуславя обмена на библиографски и контролни записи. Интеграция – описанията на всички видове материали и контролираните форми на имената (наименованията) на обектите трябва да се основават на общи правила, доколкото е възможно [1]. В същия документ, сред основните задачи и функции на каталога са посочени необходимостта той да бъде ефективен и добре действащ инструмент, да може да открива библиографските ресурси в библиотечния фонд и да ги идентифицира, т.е. да гарантира, че описаният обект отговаря на посочената характеристика или да разграничава два обекта с подобни характеристики [2]. Качеството на библиографското описание се гарантира допълнително от създаването на контролирани точки за достъп, които се създават за установените и вариантни форми на имената на обектите и осигуряват последователността, необходима за обединяване на библиографските записи за група ресурси. Коректните точки за достъп осигуряват надеждно търсене [3].

Изследователите, които се занимават както с проблемите на каталогизацията, така и с тези на автоматизирани библиотечни системи, са единодушни относно значимостта на стандартизацията. Те подчертават, че тя е ключова за унифицирането на дейностите и за стандартизирането на информационните ресурси, създавани от библиотеките [5]. Стандартизационните процеси в областта на библиографското описание започват през 1969 г. През 90-те години идеята за изграждане на обща рамка за идентификация и описание на ресурсите получава значителен тласък с публикуването на резултатите от теоретични изследвания, възложени от ИФЛА на секциите по каталогизация и класификация: „Функционални изисквания към библиографските записи“. Това води до създаването на общи принципи за каталогизация, разработени от ИФЛА [5]. През 2011 г. из-

лиза консолидираното издание на международния стандарт ISBD, който дефинира изискванията за описание и идентификация на най-често срещаните публикувани ресурси, включени в библиотечните колекции. Основната цел на тези стандарти е да осигурят съпоставима каталогизация в глобален мащаб, улеснявайки международния обмен на библиографски записи между националните библиографски агенции и в рамките на световната библиотечна общност [1].

М. Миланова, един от водещите изследователи в областта на библиотечно-информационните науки в България споделя, че в исторически план международните стандарти се създават успоредно с ускорените процеси по автоматизирането на библиотечно-библиографските дейности, както и че стандартите за библиографско описание благоприятстват развитието и изграждането на машинно четимите формати [4]. Същата изследователка споделя, че форматът MARC 21 е широко разпространен още след създаването си през 2000 г., както и че много от производителите на библиотечен софтуер предлагат използването на MARC 21 за нуждите на каталогизацията и обмена [4]. Нейните изводи, отнасящи се до паралелното развитие както на стандартите, така и на машинно четимите формати са, че развитието на тези процеси следва добрите международни практики. Потребителите на информационни продукти следва да бъдат улеснени в намирането на релевантна информация [5].

## 2. Преглед на съществуващи системи

През последните години библиотеките се ориентират към автоматизация и внедряване на нови софтуерни и хардуерни решения, които оптимизират управлението на библиотечните процеси. Тези системи предоставят ефективни механизми не само за обслужване на читателите, но и за автоматизация и улесняване на работните процеси на библиотечните специалисти. Освен това, възможностите за обучение на библиотечни служители и студенти, изучаващи библиотечно-информационни науки, значително се подобряват с помощта на тези технологии. В България, три от най-използваните системи за автоматизиране на библиотеките са: Автоматизирана система (АБ), уеб-базираната система Koha и уеб-базираната библиотечна система OpenBiblio. В таблица 1 е представена сравнителна характеристика между трите системи.

Таблица 1. Сравнителна характеристика между системите

№	Име	Koha	АБ	OpenBiblio
<b>Технически спецификации</b>				
1	Поддържана операционна система	Linux, Windows, Mac OS	Windows	Linux, Windows, Mac OS
2	Програмен език	Perl	Clipper	PHP
3	Уеб сървър	Apache	He	Apache
4	База данни	MySQL	dBASE III	MySQL
5	Разработчик	Horowhenua Library Trust	PC-TM	Дейв Стивънс
6	Уеб базиран интерфейс	Да	He	Да
7	Лиценз	GNU GPL	Годишен абонамент	GNU GPL
8	Инсталиране	Средно	Лесно	Лесно
9	Онлайн демо версия	Да	He	Да
10	Превод на български език	Да	Да	He
<b>Модули</b>				
11	Каталогизация	Да	Да	Да
12	Комплектуване	Да	Да	He

13	Библиотечно обслужване	Да	Да	Да
14	Онлайн електронен каталог	Да	Да	Да
15	Посетители	Да	Да	Да
16	Отчети	Да	Да	Да
17	Периодични издания	Да	Да	Да
18	Административни функции	Да	Да	Да
<b>Функции</b>				
19	Unicode UTF-8	Да	Да	Да
20	Опростено търсене	Да	Да	Да
21	Разширено търсене	Да	Да	Не
22	XML експорт	Да	Да	Не
23	Z39.50	Да	Не	Да
24	MARC 21 стандарт	Да	Не	Да
25	ISBD	Да	Да	Не
26	Търсеща машина Zebra	Да	Не	Не
27	Обработка и плащане на фактури	Да	Не	Не
28	Регистрация на потребители	Да	Да	Да
29	Глоби	Да	Да	Не
30	Електронен обмен на документи	Да	Да	Да

Допълнително, усъвършенстването на библиотечните системи в контекста на бързото развитие на информационните технологии може да се разгледа чрез анализ на техните технически характеристики и функции, като се акцентира върху техните предимства и ограничения. Ето и няколко примера:

- **Съвместимост с операционни системи:** Различните системи поддържат различни платформи - Koha и OpenBiblio са съвместими с Linux, Windows и Mac OS, докато AB работи само на Windows. Това подчертава необходимостта от разработка на по-гъвкави и универсални решения, които да бъдат по-лесно адаптирани към различни операционни среди.

- **Програмен език и база данни:** Koha използва Perl и MySQL, OpenBiblio - PHP и MySQL, а AB се основава на по-стария език Clipper и dBASE III. Това показва, че съществува потенциал за модернизирание на по-старите системи, за да се възползват от предимствата на съвременните езици за програмиране и бази данни, като се осигури по-голяма стабилност и функционалност.

- **Уеб сървър и интерфейс:** Koha и OpenBiblio използват Apache и предоставят уеб базиран интерфейс, което улеснява достъпа до системата от различни устройства. AB обаче няма тази функция, което ограничава нейната гъвкавост и потенциал за развитие. Съответно, модернизацията на библиотечните системи може да включва интегриране на уеб базиран интерфейс и съвременни уеб технологии.

- **Лицензи и инсталация:** Koha и OpenBiblio са с отворен код и под GNU GPL лиценз, което предоставя възможности за тяхното свободно подобряване и разширение от общността. От друга страна, AB е на годишен абонамент, което може да ограничи достъпа до нея и нейното развитие.

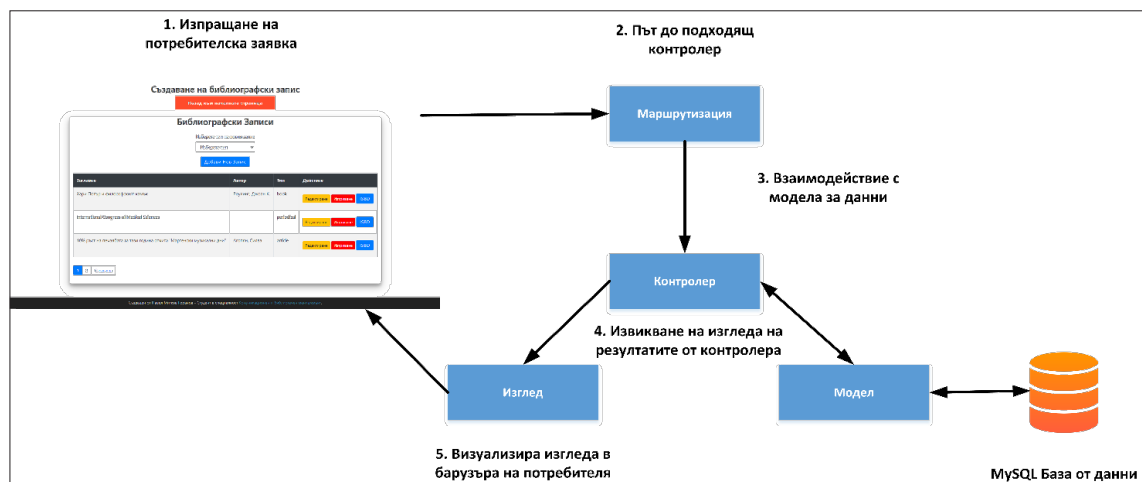
- **Модулност и функции:** Всички системи предлагат основните функции като каталогизация, комплектуване и библиотечно обслужване. Въпреки това, Koha предлага по-богат набор от функции като поддръжка на стандарта MARC 21, Z39.50 и ISBD, които не се поддържат от всички други системи. Това показва, че усъвършенстването на системи като AB и OpenBiblio може да

включва добавяне на такива стандарти и функции, за да се постигне по-голяма оперативна съвместимост и ефективност.

- Сигурност и електронни услуги: Коѝа поддър̀жа функции като обработка на фактури и плащане на глоби, което улеснява административната работа и взаимодействието с потребителите. Разширяването на тези функции в други системи би било от полза за подобряване на цялостното обслужване и управление на библиотеките.

### 3. Архитектура на уеб-базирана платформа за обучение на студенти по библиотечно-информационни науки

При разработването на уеб-базираната платформа за обучение на студенти по библиотечно-информационни науки се използва архитектурния модел: „МОДЕЛ-ИЗГЛЕД-КОНТРОЛЕР (МИК)“ – Model-View-Controller (MVC). Този архитектурен модел налага разделение между бизнес логиката от въвеждането и презентационната логика свързана с графичния потребителски интерфейс. Архитектурният модел разделя уеб-базираната система за обучение на три логически компонента: модел, изглед, контролер (фигура 1). Всеки един от тези компоненти е създаден да се справя със специфични аспекти от разработката на уеб-базираната система за обучение на студенти. МИК платформите са едни от най-често използваните за създаване на разширяващи се проекти.



**Фигура 1.** Архитектура на уеб-базирана платформа за обучение на студенти по библиотечно-информационни науки

Фигура 1 демонстрира функционирането на уеб-базирана система за обучение по библиотечно-информационни науки, използваща архитектурния модел Model-View-Controller (MVC). MVC архитектурата разделя приложението на три основни компонента – модел (Model), изглед (View) и контролер (Controller) – с цел подобряване на гъвкавостта, повторното използване на кода и опростяването на поддръжката и развитието на системата.

Процесът започва с взаимодействие между потребителя и уеб интерфейса, което в случая е визуализирано чрез форма за създаване на библиографски запис. Потребителят въвежда необходимите данни и изпраща заявката чрез потребителския интерфейс на платформата. Заявката представлява HTTP заявка, съдържаща информацията, подадена от потребителя. На този етап, системата приема потребителските данни и започва да ги обработва в съответствие с предефинираните правила на приложението. Тази стъпка е критична, тъй като точното събиране на входните данни от потребителя е първата стъпка към успешното създаване на библиографски запис в съответствие със стандартите за каталогизация.

След изпращане на заявката, тя преминава през модул за маршрутизация, който играе ключова роля в MVC архитектурата. Маршрутизаторът идентифицира правилния контролер, отговарящ за изпълнението на заявката на потребителя, като използва конфигурацията на пътищата в приложението. За всяка заявка съществува ясно дефиниран маршрут, който определя кой контролер ще обработва конкретната операция. Маршрутизацията позволява на системата да насочи правилно всяка заявка до подходящия контролер въз основа на подадените данни и заявените действия. Това гарантира, че потребителските заявки се обработват ефективно и в съответствие с логиката на приложението.

Контролерът, след като получи заявката от маршрутизатора, започва взаимодействие с модела за данни. Моделът представлява логиката за достъп и управление на данните в базата данни на приложението. В този контекст, моделът е отговорен за съхраняването на библиографските записи в MySQL базата от данни и за извличането на данни при необходимост. Моделът изпълнява бизнес логиката на приложението, като валидира и обработва данните, подадени от потребителя. В случая на уеб-базираната платформа за обучение по библиотечно-информационни науки, моделът осигурява правилното съхранение на библиографските данни в съответствие със стандарти като MARC 21 и ISBD, като гарантира консистентността и точността на описаните обекти.

След като контролерът завърши взаимодействието с модела, обработените данни се предават на изгледа (View). Изгледът отговаря за визуалното представяне на информацията и формира отговор, който ще бъде изпратен обратно към потребителя. Изгледът трансформира данните в човешко четим формат, като представя информацията по структуриран начин. В този случай, изгледът визуализира създадения или редактирания библиографски запис в удобен за потребителя интерфейс. MVC архитектурата ясно отделя изгледа от бизнес логиката и обработката на данните, което подобрява гъвкавостта на системата.

Крайната стъпка в MVC архитектурата е визуализирането на изгледа в браузъра на потребителя. Това се осъществява чрез генериране на HTML страница, която отразява резултатите от заявката, подадена от потребителя. Потребителят може да види информацията, свързана с новосъздадения или актуализирания библиографски запис, като данните са форматираны в съответствие с изискванията на системата. Тази стъпка е особено важна за потребителското изживяване, тъй като удобството на навигацията и представянето на данните играе съществена роля за ефективността на обучението и работата със системата.

Архитектурата на системата, представена на фигура 1, представлява ефективен и структуриран подход за обработка на потребителски заявки и управление на данни. MVC моделът осигурява ясно разграничение между логиката на бизнес процесите, визуализацията и достъпа до данни, което улеснява поддръжката и разширяването на системата. Модулната архитектура на приложението позволява лесно добавяне на нови функционалности и осигурява мащабируемост, необходима за бъдещото развитие на платформата.

#### **4. Логически и функционален модел на уеб-базираната платформа**

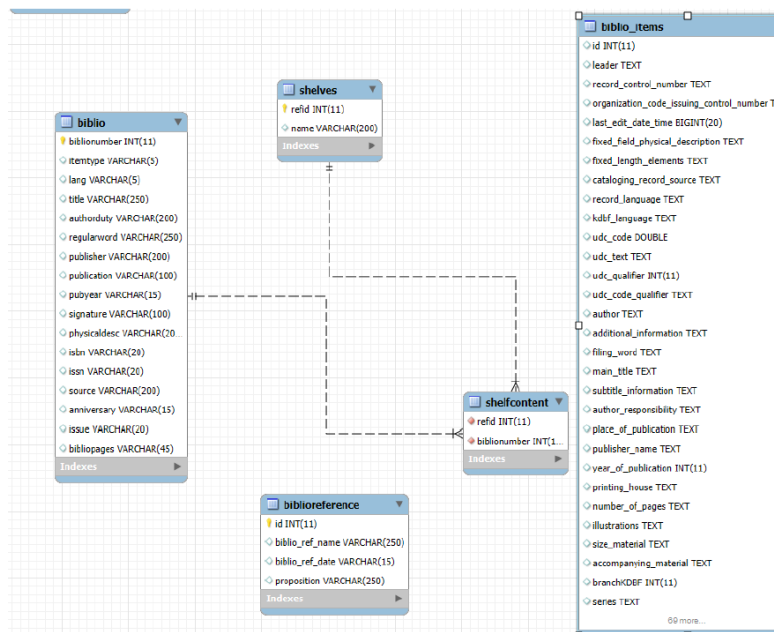
##### **4.1. Логически модел на уеб-базираната платформа**

При разработване на платформата е внедрена релационна база данни, която организира и съхранява информацията във формат на релации, съставени от записи (редове) и атрибути (колони). За ефективното управление на тази база данни е използвана релационната система за управление на бази данни (RDBMS) MySQL, която се отличава със своята производителност и надеждност при работа с големи обеми от данни. В рамките на реализацията е приложена подсистемата InnoDB, която предлага поддръжка на транзакции и гарантира целостта на данните чрез механизми като заключвания на редове и външни ключове.

Като колагия за базата е избрана `utf8_unicode_ci`, което позволява пълноценното съхранение и обработка на текстови данни на кирилица, както и на други нелатински азбуки. Това е особено важно за системи, опериращи с многоезични данни и предоставящи поддръжка за различни набори от символи, което значително разширява възможностите за интернационализация на платформата. Колагията `utf8_unicode_ci` осигурява правилно сортиране и сравняване на текстове, което е



критично за точността на заявките в базата данни, включително тези, свързани с имената на файлове, потребители и директории на кирилица и други нелатински символи. Този подход гарантира висока степен на гъвкавост и съвместимост на системата с глобални стандарти за данни, което я прави подходяща за интеграция с други платформи и системи, използващи различни езикови и текстови стандарти. На фигура 2 е представена релационната схема на базата данни.



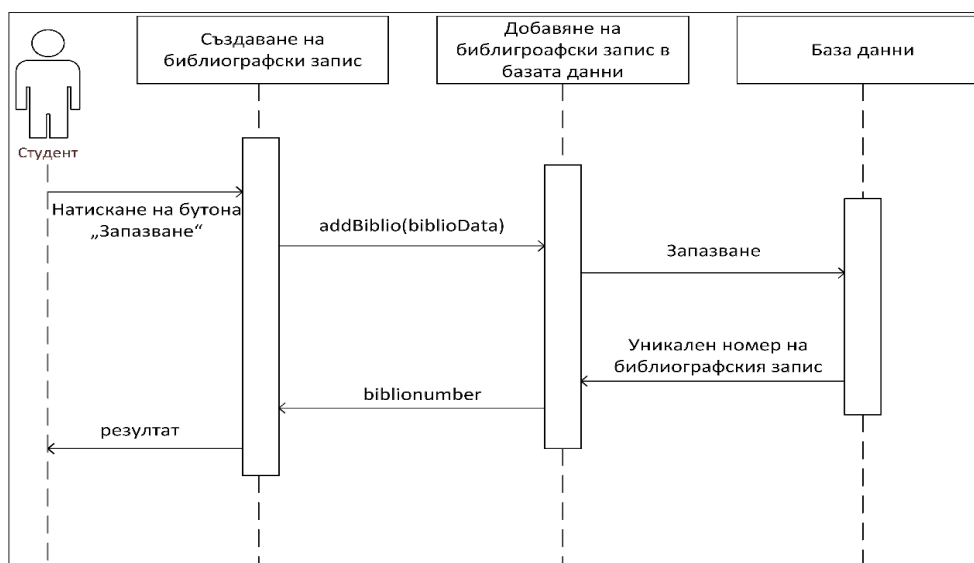
Фигура 2. Релационна схема на базата от данни

#### 4.2. Функционален модел на уеб-базираната платформа

Функционалният модел на системата представлява структурирано описание на основните функции, които тя изпълнява, както и взаимодействието между отделните компоненти. Той се фокусира върху начина, по който системата осъществява своите задачи, като същевременно показва потока на информацията и начина, по който потребителите взаимодействат с нея. Основните функции на системата са:

- **Управление на библиографски записи:** Системата позволява на потребителите да създават, редактират и изтриват библиографски записи. Всеки запис включва метаданни като автор, заглавие, година на издаване, ISBN и други библиографски данни. Тези данни се съхраняват в базата и могат да бъдат достъпвани или обновявани в реално време.
- **Автоматизирано генериране на библиографски справки:** Системата предоставя функция за автоматично създаване на библиографски справки и списъци според международно признати стандарти като MARC 21 и ISBD. Потребителите могат да генерират различни видове библиографски списъци, които отговарят на специфични изисквания или нужди.
- **Поддръжка на мултиезичност:** Благодарение на използваната кодация utf8\_unicode\_ci, системата поддържа въвеждането и обработката на данни на кирилица и други нелатински азбуки. Това осигурява лесна работа с различни езици и прави системата удобна за употреба в международен контекст.
- **Интерфейс за преглед на резултати:** След успешното въвеждане на заявка или търсене, системата предоставя на потребителя визуализация на резултатите чрез удобен за ползване интерфейс. Всяка заявка се обработва и резултатите се визуализират в подходящ формат, като се показва пълната информация за съответните записи.

На фигура 3 е представена диаграма на последователност при създаване на библиографски запис.



**Фигура 3.** Диаграма на последователност при създаване на библиографски запис

Диаграмата на последователността илюстрира процеса на създаване и добавяне на библиографски запис в базата данни от страна на потребителя, представен тук като “Студент”.

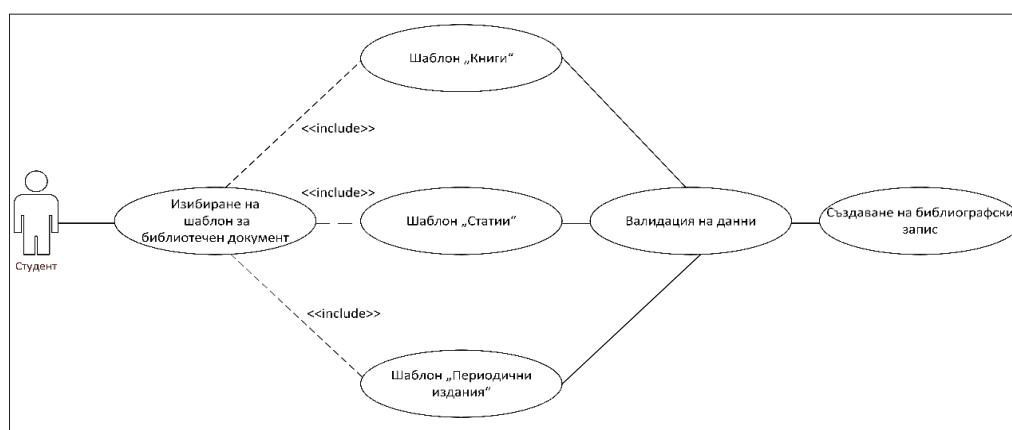
Участниците в диаграмата включват „Студент“, който представлява потребителя, и „Система“, която включва функционалностите за създаване и добавяне на библиографски запис, както и „База данни“, където се съхранява информацията за библиографските записи.

Процесът започва със създаването на библиографски запис, при което студентът влиза в интерфейса на системата и започва да попълва необходимата информация. След като попълни данните, той натиска бутона “Запазване”, който задейства процеса на запазване. Системата извиква функцията `addBiblio`, която приема информацията (`biblioData`), предоставена от студента. След успешното извикване на функцията, системата добавя новия библиографски запис в базата данни. Процесът на взаимодействие с базата данни завършва с операцията за запазване на данните.

След успешно добавяне на записа, системата връща уникален идентификатор (`bibliNumber`), който идентифицира конкретния библиографски запис. В крайна сметка студентът получава резултата от операцията, който потвърждава успешното добавяне на библиографския запис.

Тази диаграма представя основните стъпки и взаимодействия в процеса на създаване и добавяне на библиографски записи, демонстрирайки логиката и потока на данните между потребителя и системата. Тя е полезна за разбиране на функционалността на платформата и взаимодействието между различните компоненти.

На фигура 4 е представена диаграма на дейност при създаване на библиографски запис.



**Фигура 4.** Диаграма на случаите на употреба – създаване на библиографски запис



Диаграмата представя процеса на избор и създаване на библиографски запис от студент. Студентът първо избира шаблон за библиотечен документ, който може да бъде за книги, статии или периодични издания. Всеки от тези шаблони е свързан към процеса на валидиране на данните, който проверява дали въведената информация отговаря на изискванията. След успешна валидация се създава библиографски запис. Всеки от трите типа документи използва процеса на валидиране, преди да бъде генериран окончателният запис.

## **5. Имплементация на уеб-базирана платформа за обучение на студенти по библиотечно-информационни науки**

При реализацията на уеб-базираната платформа са използвани следните технологии: PHP, MySQL, HTML5, CSS3, JavaScript, JQuery, Bootstrap и W3.CSS. Причините, за използване на всяка една от изброените технологии, са описани в следващите точки.

### **5.1. Php**

PHP е скриптов език върху сървърната (обслужваща) страна [7]. Език с отворен код, който е проектиран за уеб програмиране и е широко използван за създаване на сървърни приложения и динамично уеб съдържание. PHP е рекурсивен акроним от PHP: Hypertext Preprocessor (като в самото начало има значение, дадено от създателите му, на Personal Home Page). „PHP е скриптов език със синтаксис базиран на C“ и Perl [8]. Използва се предимно в Web-среда за реализиране на широк кръг от услуги. Той е един от най-популярните езици за програмиране в Интернет и популярността му расте непрекъснато. Някои от конкретните причини за използване на Php са: активна общност и достъпност до ресурси за поддръжка, нисък разход за хостинг, гъвкавост и мащабируемост, лесна интеграция с библиотеки и системи за управление на съдържание.

### **5.2. MySQL**

MySQL е една от най-популярните системи за управление на релационни бази данни с отворен код. Тя е широко използвана за съхранение и управление на данни в уеб приложения, благодарение на своята скорост, надеждност и мащабируемост. MySQL поддържа стандартния език за заявки SQL и е съвместима с множество операционни системи, като позволява на разработчиците лесно да създават, модифицират и извличат данни от базите си данни. Тя е основен компонент в много уеб технологии, включително LAMP (Linux, Apache, MySQL, PHP/Python/Perl). Използва се за съхранение и управление на данни, като осигурява надеждно и мащабируемо управление на бази данни за обучителната платформа.

### **5.3. HTML5 и CSS3**

HTML5 е последната версия на езика за структуриране на уеб съдържание HTML (HyperText Markup Language). Тя въвежда нови елементи и атрибути, които улесняват създаването на по-интерактивни и мултимедийни уеб страници [9]. Определя структурата и съдържанието на уеб страниците, като формира основата на всяка уеб платформа. CSS3 (Cascading Style Sheets, Level 3) е последната версия на езика за стилове, използван за оформяне и поддръждане на уеб страници. CSS3 въвежда множество нови функции и подобрения, които позволяват на разработчиците да създават по-сложни и визуално привлекателни дизайни [10]. Отговаря за оформлението и стилизирането на уеб страниците, като прави интерфейса визуално привлекателен и лесен за навигация.

### **5.4. Java Script**

JavaScript е динамичен, интерпретиран програмен език, който е основен инструмент за разработка на интерактивни и динамични уеб страници. Той позволява на разработчиците да добавят функционалности като валидиране на формуляри, манипулиране на HTML и CSS, управление на мултимедия и създаване на сложни уеб приложения [11]. Позволява добавяне на интерактивност и динамично съдържание към уеб страниците, което подобрява потребителското изживяване.

### **5.5. JQuery и Bootstrap**

jQuery е популярна JavaScript библиотека, създадена с цел да опрости манипулацията на HTML документи, управлението на събития, анимациите и AJAX взаимодействията в уеб страници [12]. Bootstrap е популярна CSS фреймуърк, предназначена за бързо и лесно създаване на адаптивни и елегантни уеб сайтове и приложения. Разработена от екипа на Twitter, Bootstrap пре-

доставя готови CSS и JavaScript компоненти като навигационни барове, бутони, форми, карусели и модални прозорци, които могат лесно да се интегрират и персонализират [13]. Улеснява работата с JavaScript чрез предоставяне на опростени функции за манипулиране на HTML елементи и събития.

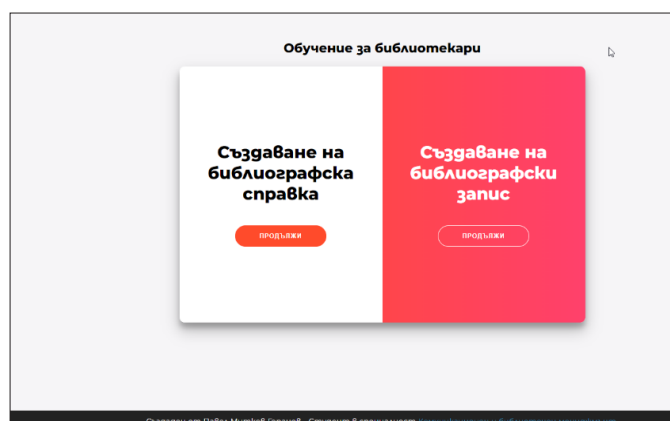
#### 5.6. PhpStorm, PhpMyAdmin и Xampp

PhpStorm е мощна интегрирана среда за разработка (IDE), създадена специално за PHP разработчици. PhpStorm предлага богата гама от инструменти и функции, които улесняват процеса на програмиране, като интуитивно автодовършване на код, интелигентно навигиране и бързо рефакториране [14].

phpMyAdmin е популярна уеб-базирана платформа, предназначена за управление на MySQL и MariaDB бази данни. Тя предоставя лесен за използване графичен интерфейс, който позволява на потребителите да извършват разнообразни задачи, свързани с базите данни, без да е необходимо да използват командния ред [15].

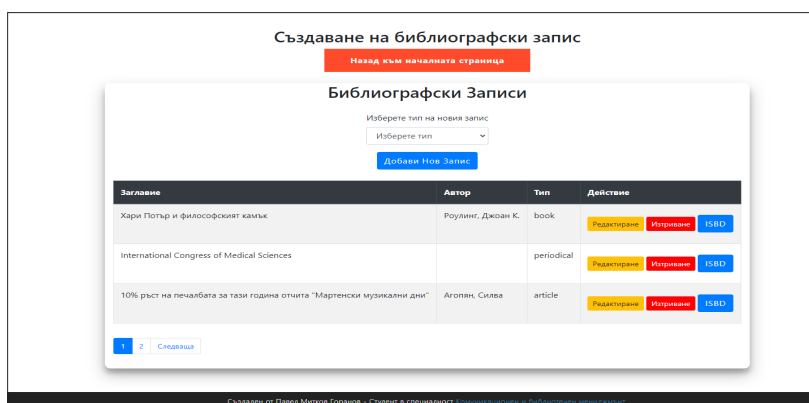
XAMPP е лесен за инсталиране и използване уеб сървър, който включва пакет от софтуерни компоненти, необходими за разработка и тестване на уеб приложения. Той съдържа Apache HTTP Server, MariaDB (или MySQL) база данни, и поддръжка за скриптови езици като PHP и Perl, което го прави цялостно решение за разработчиците. XAMPP е мултиплатформен и може да работи на различни операционни системи като Windows, macOS и Linux, предоставяйки еднаква среда за разработка на различни устройства [16].

На фигура 5 е представен началният екран на уеб-базираната платформа за обучение на студенти по библиотечно-информационни науки.



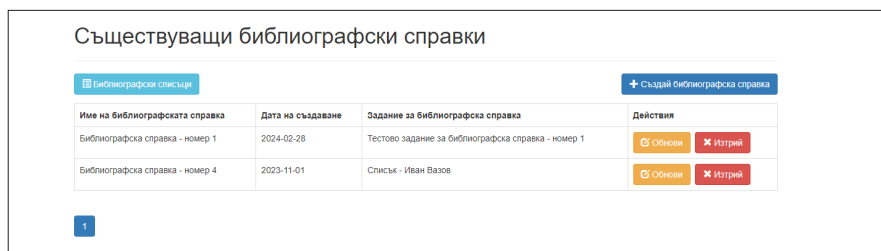
Фигура 5. Начален екран на уеб-базираната платформа за обучение на студенти по библиотечно-информационни науки

На фигура 6 е представен модула за създаване и управление на библиографски записи.



Фигура 6. Начален екран на модула за създаване и управление на библиографски записи

На Фигура 7 е представен модула за създаване и управление на библиографски справки и списъци.



Фигура 7. Начален екран на модула за създаване и управление на библиографски справки и списъци

На фигура 8 е представен шаблон за създаване на библиографски запис от тип „Книга“.

**MARC 21 Шаблон за Книги**

0 1 2 3 4 5 6 7 8 9

Запази

**Раздел 0**

000 ? - Лидер  
00 Лидер [лат a22 а 4500] [Задължително](#)

001 ? - Контролен номер на записа

003 ? - Код на организацията, даваща контролен номер  
00 Код на организацията, даваща контролен номер [BLK] [Задължително](#)

005 ? - Дата и час на последна редакция  
00 Дата и час на последна редакция [ ] [Задължително](#)

007 ? - Фиксирано поле за физическо описание  
00 Физическо описание [a] [Задължително](#)

008 ? - Елементи от данни с фиксирана дължина  
00 Елементи с фиксирана дължина [22104b bu ||||| 00 | bu d] [Задължително](#)

020 ? - Международен стандартен номер на книга ISBN  
a ISBN [ ] [Задължително](#)

q Коментар към ISBN [ ]

040 ? - Източник на каталожния запис  
a Код на библиотеката [BLK] [Задължително](#)

041 ? - Код за език  
a Език [Албански] [Задължително](#)

с Език за КДБФ [Български] [Задължително](#)

080 ? - УДК  
a УДК текст [ ] [Задължително](#)

b УДК код [ ] [Задължително](#)

Фигура 8. Шаблон за създаване на библиографски запис от тип „Книга“

На фигура 9 е представен ISBD изглед на библиографски запис от тип „Книга“.

**ISBD изглед**

Роулинг, Джоан К.  
Хари Потър и философският камък / Джоан К. Роулинг ; Прев. Теодора Жебарова. - София : Егмонт България, 2017. - 280 с. 23 см.  
ISBN 978-954-446-468-4

821.111-93+

[Затвори](#)

Изберете тип

Фигура 9. ISBD изглед на библиографски запис от тип „Книга“

## ЗАКЛЮЧЕНИЕ И БЪДЕЩА РАБОТА

В заключение, интегрираната уеб-базирана платформа за обучение на студенти по библиотечно-информационни науки представя иновативно решение за управление на библиографски записи, което отговаря на съвременните изисквания за достъпност и ефективност. Чрез създаването на система, базирана на международно признати стандарти като MARC 21 и ISBD, платформата

предлага удобен интерфейс за потребителите, което улеснява както създаването, така и управлението на библиографски ресурси.

Досегашната работа по системата показва значителен напредък в областта на библиотечните технологии, като осигурява необходимите инструменти за студентите да се запознаят с процесите на каталогизация и управление на информация. Платформата не само отговаря на нуждите на потребителите, но и подготвя студентите за предизвикателствата на съвременните библиотечни практики.

За бъдещата работа е необходимо да се осигури продължаваща поддръжка и актуализация на системата, както и разширяване на функционалността, за да отговори на новите изисквания и тенденции в библиотечното дело. Планираме да интегрираме нови модули, които да предлагат допълнителни функции за анализ на данни и отчетност, което ще повиши ценността на платформата за потребителите.

Също така, в бъдеще ще бъде важно да се извършат проучвания и оценки на удовлетвореността на потребителите, за да се идентифицират области за подобрене и оптимизация на потребителския опит. Освен това, разширяването на платформата за включване на международни библиографски стандарти и интеграция с други библиотечни системи ще улесни международния обмен на библиографски данни, което е от съществено значение за глобализацията на библиотечните услуги.

В заключение, платформата представлява важна стъпка напред в съвременната библиотечна практика и с продължаващото ѝ развитие, тя има потенциала да бъде ценен ресурс за бъдещите поколения библиотекари.

## ЛИТЕРАТУРА

[1.] [Ай Ес Би Ди] ISBD: Международен стандарт за библиографско описание: Консолидирано издание: Препоръчано от Групата за преглед на ISBD и одобрено от Постоянната комисия на Секцията по каталогизация на ИФЛА / Прев. от англ. А. Дипчикова ; Ред. В. Людсканова. София: ББИА, 2012.

[2.] Изложение на международните принципи на каталогизацията: Въведение. ИФЛА. 2009 1.07.2024.

[3.] **Миланова, Милена**. Българската каталогизация в глобалното информационно пространство на XXI век. Анализи, стратегии, перспективи, СУ „Св. Климент Охридски“. Филос. фак. Кат. Библиотекознание, научна информация и културна политика, [Дис. за образователна и научна степен доктор], Утвърдена от ВАК с протокол N. 12–6 / 18.05.2009. София, 291 л.: с табл.

[4.] **Яврукова, Биляна**. Нови стандарти в областта на библиотечно-информационните науки. // ББИА онлайн, 11, 2021, № 2, с. 19–23.

[5.] **Янакиева Татяна**. Универсалният език на стандартите. Предизвикателствата пред каталогизацията в България през XXI век. // Годишник на Софийския университет „Св. Климент Охридски“. Фило-софски факултет. Книга библиотечно-информационни науки, Т. 4, 2012, с. 47–61.

[6.] **Atkinson, Leon** and Zeev Suraski. Core PHP Programming. Prentice Hall, 2004.

[7.] Bootstrap – <https://bg.wikipedia.org/wiki/Bootstrap> (Ползван на 26.09.2024 г.)

[8.] **Flanagan, David**. JavaScript: The Definitive Guide. O'Reilly Media, Inc, 2006.

[9.] **Golemon, Sara**. Extending and Embedding PHP. Sams Publishing, 2006.

[10.] **Howe, Shay**. Learn to Code HTML and CSS: Develop and Style Websites. New Riders, 2014. JQuery – <https://bg.wikipedia.org/wiki/JQuery> (Ползван на 26.09.2024 г.)

[11.] **Meyer, Eric A.** CSS: The Definitive Guide. O'Reilly Media, Inc, 2007. phpMyAdmin 2024 – <https://www.phpmyadmin.net/> (Ползван на 26.09.2024 г.)

[12.] PhpStorm 2024 – <https://www.jetbrains.com/phpstorm/> (Ползван на 26.09.2024 г.)

[13.] XAMPP 2024 – <https://bg.wikipedia.org/wiki/XAMPP> (Ползван на 26.09.2024 г.)

## ИНФОРМАЦИЯ ЗА АВТОРА

**Павел Митков Горанов**, Системен администратор, Регионална библиотека „Любен Каравелов“ – Русе, E-mail: [PavelGoranov@gmail.com](mailto:PavelGoranov@gmail.com).

## ABOUT THE AUTHOR

**Pavel Mitkov Goranov**, System Administrator, Lyuben Karavelov Regional Library – Ruse, E-mail: [PavelGoranov@gmail.com](mailto:PavelGoranov@gmail.com).