**Księgarnia Internetowa** – (Online Bookstore Web Application) – **Wymagania Funkcjonalne** aplikacji

----------------------------------------------------------------------------------------------------------------------------------------------------

**Wymagania Funkcjonalne** –

~~-~~ *~~Zachowanie Aplikacji w określonych warunkach~~*~~,~~ *~~Cechy Aplikacji~~*~~,~~ *~~Funkcje Aplikacji~~*

- **Szczegóły techniczne**, **Przetwarzanie Danych**, **Uwierzytelnianie**

„**Opisują co dana aplikacja powinna robić**, odnoszą się do **konkretnych funkcjonalności**”

*Przykład :*



-----------------------------------------------------------------------

**User Stories** – *Historyjki użytkownika* - **cechy** i **właściwości systemu** *z punktu widzenia użytkownika*,

„*As a [user], I want [what?] so that [reason/value]*”.

Takie podejście ułatwi zrozumienie potrzeb potencjalnego **użytkownika** aplikacji.

np.: „ Jako administrator, chcę mieć możliwość generowania **raportów** z systemu, aby wiedzieć, ilu użytkowników odwiedza moją stronę. „

w odniesieniu do tego, przykład **Kryteriów akceptacji : - Lista kryteriów, która stwierdza, czy dana funkcjonalność jest spełniona (zrealizowana) :**

*- Raport może generować administrator,*

*- Raport powinien składać się z ID użytkownika, daty, nazwy odwiedzanej strony, ilości czasu spędzonego na danej stronie,*

*- Raporty powinny generować się po kliknięciu przycisku.*



`







----------------------------------------------------------------------------------------------------------------------------------------------------

**Wymagania Funkcjonalne**

**Do zrobienia**

* System logowania.

|  |  |
| --- | --- |
| Z punktu widzenia **użytkownika** : | Z punktu widzenia aplikacji / **systemu** : |
| - Użytkownik może zalogować się na swoje konto.  - Użytkownik może wylogować się ze swojego konta  - Logowanie za pomocą **e-mail**, **telefonu** ? | - Aplikacja sprawdza czy istnieje konto o takiej nazwie użytkownika,  - W przypadku podania istniejącego loginu, następuje weryfikacja tego, czy użytkownik podał poprawne hasło.  - W przypadku podania prawidłowych danych dostępowych, następuje pobranie danych użytkownika, umieszczenie ich w zmiennych sesyjnych oraz przekierowanie na stronę główną. Użytkownik od teraz ma dostęp do swojego konta.  - Poprawa implementacji logowania w .php ->  Wydzielenie do osobnych plików, użycie functions.php … itd. |

*Kryteria akceptacji :*

*- „Użytkownik może zalogować się na swoje konto tylko w przypadku podania prawidłowych danych swojego konta „ …*

|  |  |
| --- | --- |
| Z punktu widzenia **użytkownika** : | Z punktu widzenia aplikacji / **systemu** : |
| - Użytkownik może zalogować się na swoje konto.  - Użytkownik może wylogować się ze swojego konta  - Logowanie za pomocą **e-mail**, **telefonu** ? | - Aplikacja sprawdza czy istnieje konto o takiej nazwie użytkownika,  - W przypadku podania istniejącego loginu, następuje weryfikacja tego, czy użytkownik podał poprawne hasło.  - W przypadku podania prawidłowych danych dostępowych, następuje pobranie danych użytkownika, umieszczenie ich w zmiennych sesyjnych oraz przekierowanie na stronę główną. Użytkownik od teraz ma dostęp do swojego konta. |

**------------------------------------------------------------------------------------------------------------------------------------------------**

Online bookstore - **Wymagania funkcjonalne** (Funkcjonalności)

**Ważne** na **żółto Mniej ważne** na niebiesko

------------------------------------------------------------------------------------------------------------------------------------------------

* **System logowania**. Użytkownik może **zalogować się** na swoje konto

**- Do Zweryfikowania** (Logowanie w PHP).

- Logowanie za pomocą **e-mail**~~,~~ **~~telefonu~~** ~~?~~

Design: Wystylizować zaloguj.php

~~- dodanie informajci o tym, żę włączony jest Caps-Lock przy logowaniu.~~

- Sprawdzenie [If] - czy jesteśmy zalogowani - w kazdym pliku, do którego ma dostęp tylko zalogowany użytkownik.

---------------------------------------------------------------------------------------------------------------

* **System rejestracji** nowego konta użytkownika. Użytkownik może stworzyć konto
* **Walidacja i sanityzacja danych rejestracji (login, hasło, email …)**

~~-> walidacja daty -> php4 checkdate(m,d,r)~~

* **reCaptcha**

-> (Ewentualnie!) Wysyłanie na **email kodu weryfikacyjnego** po rejestracji (utworzeniu konta) w celu **potwierdzenia**

* Dodanie opcji „**Przypomnij hasło**” przy **logowaniu** + *wysyłanie kodu (nowego hasła na email ?) na email (wysyłanie email przez php) (?)*

wysyłanie maili (php) -> <https://phppot.com/php/sendmail-in-php-using-mail-smtp-with-phpmailer/> // Zelent -> php5, php6.

Dodanie opcji o "sile hasła" przy rejestracji. (JS ?)

- Mniej danych przy rejestracji - tylko email i hasło ? - Dodatkowe dane można uzupełnić w proiflu użytkownika.

---------------------------------------------------------------------------------------------------------------

* **Zarządzanie kontami użytkowników**
* Użytkownik może **zmienić swoje dane** (dane swojego konta) (**Do Zweryfikowania**) - Nie powinno się przechowywać **haseł** w **sesji** ! - conajmniej **id** zalogowanego użytkownika.

WYBRAĆ TE DANE, KTÓRE CHCĘ ZMIENIAĆ, NIE ZMIENIAĆ WSZYSTKICH !

(Zastanowić się, jakie (tzn **które**) dane edytować ?),

- które dane wyświetlić?

* Użytkownik może usunąć swoje konto
* …
* ~ Walidacja (PHP) **formularza edytowania danych** - czy email to email ?, pesel, kod pocztowy itd ...
* WALIDACJA - zarówno PHP (backend) jak i JS (frontend) !

---------------------------------------------------------------------------------------------------------------

* Wyszukiwarka. Użytkownik może wyszukiwać książki po nazwie (**tytule**)

po autorze

według wybranych **kategorii**

- **Walidacja** danych wprowadzonych z wyszukiwarki + **Sanityzacja**

(sprintf, myreal\_escape\_strings)

Walidacja wyszukiwarki - danych wprowadzonych od usera. **(?)**

~~Przycisk Enter = wyszukiwanie,~~ **~~ESC~~**~~,~~ **~~usuwanie~~** ~~?~~

-> wyszukiwarka - **historia wyszukiwania**. Okienko wyskakujące przy polu wyszukiwania



---------------------------------------------------------------------------------------------------------------

* Koszyk Użytkownik może **dodać** książki do koszyka

~~(nie można dodać nowej tej samej książki)~~

**-> po dodaniu istniejącej książki, następuje aktualizacja jej ilości (w koszyku)**

**- może usunąć książki z koszyka**

**- może edytować liczbę egzemplarzy - jQuery?**

**- edycja pola (input) przechowującego liczbę egzemplarzy - po kliknięciu (+) lub (-),**

(**cena zmienia się dynamicznie**)

**następnie po kliknięciu "Złóż zamówienie" - ilość jest pobierana z tego pola.**

~~Coś jest nie tak z koszykiem, nie można dodać zamówienia~~



JEŚLI KOSZYK JEST PUSTY - NIE POWINNO DAĆ SIĘ ZŁOŻYĆ ZAMÓWIENIA !

---------------------------------------------------------------------------------------------------------------

* **Zamówienie** (przejście do tej strony następuje z koszyka**,** jak i z panelu użytkownika)

- Podczas realizacji zamówienia, wyświetlają się dane z koszyka

- Użytkownik może anulować zamówienie

- Użytkownik może wyświetlać **zrealizowane** zamówienia (*backcend: dane z trzech tabel*) - ~~SQL - funkcje grupujące? - MOŻE NIEKONIECZNIE ?~~

(Problem) : Bezpośrednio po dodaniu książki do zamówienia, nie wyświetla się ona w koszyku (na stronie zamówienia - koszyk.php)

- *Edytowanie zamówienia ? (RESEARCH) -*

-> Użycie Date (php) - w jakiś sposób - przy zamówieniach ? - **(Do Zweryfikowania)**

- ~~Użycie Daty przez php przy wstawianiu zamówienia (INSERT),~~

czy

**użycie ~~TIMESTAMP~~ DATETIME przy dodaniu zamówienia (INSERT) - przy zapytaniu insert, data złożenia zamówienia, powinna miec typ TIMESTAMP - wtedy jej wartość będzie równa czasowi, w którym wykonano zapytanie.** ?

Pytanie: Zamówienie realizować tylko dla zalogowanych użytkowników czy także dla użytkowników bez konta ? - Możliwość zakupu (koszyk) bez zakładania konta

- Możliwość Zrealizowania zamówienia bez zakładania konta

**Możliwość zakupu (koszyk) bez zakładania konta**

---------------------------------------------------------------------------------------------------------------

Koszyk / Zamówienia - **Co zrealizowałem w aplikacji** :

- **Dodając** książke(i) do koszyka, wstawiany jest rekord do tabeli **koszyk**

- Realizując zamówienie, dodawane są rekordy do tabeli: **zamówienia**, **płatności**,

**szczegóły zamówienia** (**insert na podstawie tabeli koszyk !**)

---------------------------------------------------------------------------------------------------------------

**Filtrowanie** - (blok nav) - JS (?) / jQuery ? / PHP? - filtrowanie zawartości content (książki)

autor cena rok wydania wydawnictwo status (dostępna/niedostępna)

**Filtrowanie - zaawansowane** - ! Patrz research ! - Jak wygląda na innych stronach ?

(**Baza Danych**) : Dodanie **podkategorii** do bazy danych tj .: Informatyka, Bazy danych

(ma to być nowa kolumna tabeli **książki** - **Podkategoria**)

---------------------------------------------------------------------------------------------------------------

Wyświetlanie książek !

- Po kliknięcie na daną książkę - wyświetlana jest informacja o tym **czy jest dostępna** (na podstawie tabeli i relacji z nią - **magazyn\_książki**).

~~Dodanie statusu ksiażki "Dostępna, niedostępna" - (Magazyn)~~  - Nie będę tego robił w BD, tylko wyświetlę tę informację, na podstawie tabeli **magazyn\_ksiazki.**

(jeśli w tabeli magazyn-ksiazki, ilosc danego egzamplarza wynosi **zero**, to ksiazka jest **niedostepna**) (to jest przykładowe rozwiązanie).

Książki -> **Komentarze** (?) / Lub dodanie **opinii** w skali oceny (5/5).

Ocena książki - na podstawie ocen użytkowników

-> Co umieścić na stronie głównej ? (index.php)

Bestsellery -> książki które sprzedałysię w więcej niż (X) sztuk ...

-> "One Page Design"

---------------------------------------------------------------------------------------------------------------

**Obsługa zamówień (?) przez personel + Przeglądane stanów magazynowych - ~~RESEARCH !~~**

(film na yt)

Panel admina - jakie **operacje** może realizowac administrator ?

- **Przeglądanie kont użytkowników**

- Przeglądanie **zamówień** - **realizacja** zamówień (**zmiana** stanu/**statusu** zamówienia)

- Przeglądanie stanów magazynowych - ...

SELECT m.id\_magazynu, m.id\_ksiazki, m.ilosc\_dostepnych\_egzemplarzy, k.id\_ksiazki, k.tytul, k.cena, k.rok\_wydania

FROM magazyn\_ksiazki AS m, ksiazki AS k

WHERE m.id\_magazynu=1 AND m.id\_ksiazki = k.id\_ksiazki;

- Dodawanie książek

- Dodawanie zdjęcia książki

- Edytowanie książek

! **Generowanie pdf** - z podsumowaniem zamówienia (ala "faktura")

---------------------------------------------------------------------------------------------------------------

~~! Wyodrębnić połączenie z bazą danych - jako osobną funkcję, przyjmującą jako parametr zapytanie sql (query) !~~

---------------------------------------------------------------------------------------------------------------

Strona ma być responsywna

(Bootstrap / CSS ?) -> Patrz AI - któreś lab (na moim git)

-> Zelent / Bootstrap

---------------------------------------------------------------------------------------------------------------

**-> Zastosowanie STRON ("Paginacja") (poprzednia, następna -> [1], [2], [3], ... ) w bloku content ...**

-------------------------------------------------------------------------------------------------------------------

implementacja przyjaznych adresów (xampp) (odcinek któryś z html/css - chyba html4)

Użycie znaczników HTML5 (?)

-------------------------------------------------------------------------------------------------------------------

**Uzupełnienie bazy danych danymi.**

! **Zmiana na odpowienie typy danych** w bazie danych !

-------------------------------------------------------------------------------------------------------------------

-------------------------------------------------------------------------------------------------------------------

(Praca dyplomowa) -> SQL Injection (zabezpieczenie przed), XSS (zabezpieczenie, jak zrealizowano)