PYTANIE 1.

Tabela 1: Ksiazki

- KsiazkaID (KP): Klucz podstawowy, unikalny identyfikator każdej książki w systemie.
- Tytul: Tytuł książki.
- Autor: Autor książki.
- Rok_wydania: Rok wydania książki.
- Stan: Status książki, przyjmuje wartości "Dostępna" lub "Wypożyczona",w zaleznosci czy ksiazka jest dostepna czy wypozyczona.

Tabela 2: Wypozyczenia

- WypozyczenieID (KP): Klucz podstawowy, unikalny identyfikator każdego wypożyczenia.
- KsiazkaID (KO do tabela ksiazki): Klucz obcy odnoszący się do KsiazkaID w tabeli Ksiazki, identyfikuje książkę, która została wypożyczona.
- CzytelnikID (KO do tabeli uzytkownicy): Klucz obcy odnoszący się do UzytkownikID w tabeli Uzytkownicy, identyfikuje użytkownika, który wypożyczył książkę.
- Data_wypozyczenia: Data, kiedy książka została wypożyczona.
- Data_zwrotu: Data, kiedy książka została zwrócona; NULL, jeśli książka nie została jeszcze zwrócona.

Tabela 3: Uzytkownicy

- UzytkownikID (KP): Klucz podstawowy, unikalny identyfikator każdego użytkownika w systemie.
- Imie: Imię użytkownika.
- Nazwisko: Nazwisko użytkownika.
- Numer_telefonu: Numer telefonu użytkownika.
- Login: Nazwa użytkownika, służąca do logowania się do systemu.
- Haslo: Hasło użytkownika, zahashowane za pomocą MD5.
- Rola: Rola użytkownika w systemie; przyjmuje 2 wartości "Admin" lub "Uzytkownik", określające uprawnienia użytkownika.

PYTANIE 2.

Przewidziano dwie role użytkowników: Admin oraz Użytkownik. Mają one różne uprawnienia, co wpływa na możliwości dostępu i działania w systemie.

Rola "Admin"

Admin posiada maksymalny zakres uprawnień, takich jak:

Zarządzanie książkami: Admin może dodawać nowe książki do systemu, edytować informacje o istniejących książkach oraz usuwać książki.

Zarządzanie wypożyczeniami: Admin ma dostęp do historii wypożyczeń, może przeglądać aktywne jak i nie nieaktwyne wypożyczenia oraz zarządzać procesem zwrotów.

Zarządzanie użytkownikami: Admin może wyświetlić wszystkich użytkowników w bazie, tworzyć nowe konta użytkowników, edytować informacje o istniejących użytkownikach (również zmieniać ich role) oraz usuwać konta użytkowników.

Rola "Użytkownik"

Użytkownik ma dostęp ograniczony dostęp tylko do niezbędnych funkcji takich jaky:

Przeglądanie katalogu książek: Użytkownik może przeglądać dostępne książki.

Wypożyczanie książek: Użytkownik ma możliwość wypożyczenia dostępnych książek.

Zarządzanie swoimi wypożyczeniami: Użytkownik może przeglądać historię swoich wypożyczeń oraz sprawdzać status aktualnych wypożyczeni i daty ich zwrotów.

W skrócie:

Główna różnica w dostępie między rolą "Admin" a "Użytkownik" polega na poziomie kontroli nad systemem. Administratorzy mają pełną kontrolę nad zarządzaniem zasobami biblioteki i użytkownikami, a zwykli użytkownicy mają dostęp wyłącznie do funkcji związanych z korzystaniem z zasobów biblioteki.

PYTANIE 3.

Logowanie użytkownika:

SESJE:

Przesłanie danych logowania: Użytkownik wprowadza swoje dane logowania (login i hasło) na stronie logowania (login.html) i wysyła je do skryptu PHP (process_login.php) za pomocą metody POST.

Weryfikacja danych: Skrypt process_login.php odbiera dane logowania, a następnie weryfikuje je, porównując z danymi w bazie danych.

Inicjacja sesji: Jeśli dane logowania są poprawne, skrypt inicjuje sesję za pomocą funkcji session_start(). Ta funkcja tworzy unikalny identyfikator sesji, który jest przechowywany po stronie klienta w ciasteczku.

Przechowywanie danych w sesji: Skrypt następnie przechowuje w sesji istotne informacje o użytkowniku, takie jak jego ID, login, i rolę. Dzięki temu w dalszej części użytkownik jest rozpoznawany jako zalogowany, a aplikacja może dostosować dostępne opcje i interfejs do jego roli.

Utrzymanie stanu zalogowanego użytkownika I Sesja na poszczególnych stronach: Na każdej stronie, która wymaga od użytkownika bycia zalogowanym (np. admin.php, user.php), skrypt rozpoczyna się od wywołania session_start(). Funkcja odczytuje identyfikator sesji z ciasteczka przeglądarki i na tej podstawie odnajduje odpowiednią sesję.

Sprawdzanie uprawnień: Strony wykorzystują informacje zapisane w sesji, aby sprawdzić, czy użytkownik jest zalogowany oraz jakie ma uprawnienia. Na tej podstawie decydują, czy użytkownik ma dostęp do

danej strony czy też powinien zostać przekierowany na stronę logowania lub otrzymać komunikat o błędzie.

Wylogowanie: Kiedy użytkownik chce się wylogować (logout.php), skrypt usuwa dane sesji za pomocą session_destroy() i opcjonalnie czyści ciasteczko sesji, co uniemożliwia dalszy dostęp do zasobów wymagających autoryzacji bez ponownego zalogowania.

Ciasteczka

Wykorzystanie ciasteczek w autoryzacji: Identyfikator sesji przechowywany w ciasteczku PHPSESSID pozwala serwerowi na odnalezienie i przywrócenie danych sesji przechowywanych po stronie serwera. Dzięki temu aplikacja może rozpoznać zalogowanego użytkownika i zapewnić mu dostęp do zasobów zgodnie z jego uprawnieniami.

Akceptacja ciasteczek:

skrypt accept_cookies.php, który służy do obsługi zgody użytkownika na używanie ciasteczek. Kiedy użytkownik odwiedza stronę i akceptuje ciasteczka, skrypt ustawia dodatkowe ciasteczko (o nazwie cookies_accepted), które informuje aplikację, że użytkownik zgodził się na ich użycie.

PYTANIE 4.

Troche już to opisałem w pytaniu 2. Tutaj dam więcej szczeółów.

W panelu administracyjnym dostępne są funkcje, które umożliwiają zarządzanie:

- **Książkami**
- **Wypożyczeniami**
- **Użytkownikami**

jak są one zaimplementowane:

Dodawanie książek:

- Formularz dodawania książek: W panelu administracyjnym znajduje się formularz (add_book_form.php), który pozwala na wprowadzenie danych nowej książki, takich jak tytuł, autor, rok wydania oraz stan (dostępna/wypożyczona).

- Skrypt obsługujący dodawanie: Po wypełnieniu formularza i wysłaniu go, dane są przekazywane do skryptu PHP (add_book.php), który odpowiada za dodanie nowej książki do bazy danych. Skrypt ten wykorzystuje funkcje mysqli_prepare i mysqli_stmt_bind_param do stworzenia bezpiecznego zapytania SQL, które dodaje nowy rekord do tabeli Ksiazki.

Edycja książek:

- Wyświetlanie istniejących książek: Panel administracyjny umożliwia przeglądanie listy wszystkich książek wraz z opcją edycji każdej z nich. Przy każdej książce dostępny jest przycisk do edycji (edit_book.php).

- Formularz edycji książek: Po wybraniu opcji edycji, administrator jest przekierowywany do formularza edycji, który jest wypełniony aktualnymi danymi edytowanej książki. Po dokonaniu zmian, formularz jest wysyłany, a dane są przekazywane do skryptu aktualizującego rekord w bazie danych.
- Skrypt obsługujący update ksiazki: Skrypt (update_book.php) odbiera dane z formularza i aktualizuje odpowiedni rekord w tabeli Ksiazki, korzystając z zabezpieczonych zapytań SQL m.in. tych co wymieniłem w add_book.php.

Usuwanie książek

- Opcja usuwania: Obok opcji edycji, przy każdej książce w panelu administracyjnym dostępna jest również opcja usunięcia. realizowana jest przez przycisk, który wysyła żądanie do skryptu PHP odpowiedzialnego za usunięcie książki (delete_book.php)

- Skrypt usuwający książki: Skrypt ten odbiera identyfikator książki do usunięcia, a następnie wykonuje zapytanie SQL, które usuwa rekord z bazy danych. Dla bezpieczeństwa, również tutaj użyłem zapytania mysqli_prepare.

PYTANIE 5.

aby zapewnić bezpieczeństwo danych i skuteczne uwierzytelnianie użytkowników. Zaimplementowałem:

1. **Hashowanie haseł**

   Do przechowania haseł w bazie użyłem funkcji haszującej MD5, aby zabezpieczyć hasła przed odczytem przez nieupoważnione osoby.

2. **Sesje użytkowników**

   Sesje są używane do śledzenia zalogowanych użytkowników poprzez unikalny identyfikator sesji. zapobiega konieczności ponownego wprowadzania danych logowania przy każdym żądaniu.

3. **Przygotowane zapytania (Prepared Statements)**

   Tak jak wymieniłem wcześniej już tę funkcję. Aby zapobiec atakom SQL Injection, system wykorzystuje przygotowane zapytania przy interakcjach z bazą danych. Przygotowane zapytania zapewniają, że wszelkie dane wprowadzane przez użytkownika są traktowane jako parametry, a nie jako część zapytania SQL, co minimalizuje ryzyko wykonania złośliwego kodu SQL.

4. **Ograniczenie dostępu do stron na podstawie roli**

Dostęp do określonych sekcji aplikacji jest ograniczany na podstawie roli użytkownika. Na przykład, tylko użytkownicy z rolą Admin mają dostęp do panelu administracyjnego. System sprawdza rolę użytkownika przy każdym żądaniu.

5. **Ochrona przed XSS**

   Aby zapobiec atakom typu XSS, dane wprowadzane przez użytkownika są filtrowane i czyszczone przed wyświetleniem na stronie. Użyłem do tego funkcji htmlspecialchars() do neutralizacji potencjalnie niebezpiecznych znaków.

PYTANIE 6.

W pliku .rar poza projektem znajduje się dokument Word (**Instrukcja dla Użytkownika.docx**)który zawiera instrukcje dla użytkownika dotyczące korzystania z aplikacji. w tym:

- **dane dostępowe dla różnych ról**
- **Informacje o imporcie bazy danych**
- **uruchamianiu aplikacji**
- **logowaniu do systemu**
- **oraz szczegółowe instrukcje dla paneli administracyjnego i użytkownika**

Opisane są kroki do zarządzania książkami, użytkownikami, wypożyczeniami, wypożyczaniem książek, przeglądaniem własnego profilu i wylogowywaniem. Dodatkowo, dokument zawiera zrzuty ekranu ilustrujące funkcje aplikacji.

PYTANIE 7.

Proces wypożyczenia książek przez użytkowników zaimplementowano w następujący sposób:

Przeglądanie dostępnych książek: Użytkownik, po zalogowaniu się do systemu, ma możliwość przejścia do sekcji "Moje Wypożyczenia" (my_loans.php), gdzie jest lista dostępnych książek do wypożyczenia. Lista ta jest generowana na podstawie zapytania do bazy danych, które wybiera książki o stanie "Dostępna".

Wybór książki do wypożyczenia: Obok każdej dostępnej książki znajduje się „przycisk” Wypożycz, który umożliwia rozpoczęcie procesu wypożyczenia książki. Link ten kieruje do skryptu (add_loan.php) z parametrami GET, w tym identyfikatorem książki (book_id) oraz identyfikatorem czytelnika który jest pobierany z sesji użytkownika.

Proces wypożyczenia książki:

W skrypcie add_loan.php sprawdzane jest, czy użytkownik jest zalogowany (i ma prawo do wypożyczenia książki). Jeśli użytkownik nie jest zalogowany, jest przekierowywany do strony logowania (login.html).

Jeśli użytkownik jest zalogowany, skrypt pobiera identyfikator książki oraz identyfikator użytkownika z sesji.

Następnie skrypt dokonuje wpisu do bazy danych w tabeli Wypozyczenia, dodając nowy rekord wypożyczenia z aktualną datą wypożyczenia i ustawioną wartością NULL dla daty zwrotu, co oznacza, że książka jest wypożyczona, ale jeszcze nie zwrócona.

Po dodaniu rekordu wypożyczenia, skrypt aktualizuje stan książki w tabeli Ksiazki na "Wypożyczona", aby odzwierciedlić, że książka nie jest już dostępna do wypożyczenia przez innych użytkowników.

Na koniec użytkownik jest przekierowywany do strony "Moje Wypożyczenia" (my_loans.php), gdzie może zobaczyć swoje aktualne i przeszłe wypożyczenia.

Komunikacja z bazą danych: Operacje na bazie danych realizowane są za pomocą zapytań SQL przygotowywanych i wykonywanych z użyciem funkcji mysqli_prepare() i mysqli_stmt_bind_param(), co zapewnia bezpieczeństwo przed atakami SQL Injection.

Obsługa błędów: W przypadku wystąpienia błędu podczas procesu wypożyczenia (np. problemu z zapytaniem SQL) skrypt wyświetla odpowiedni komunikat błędu.