

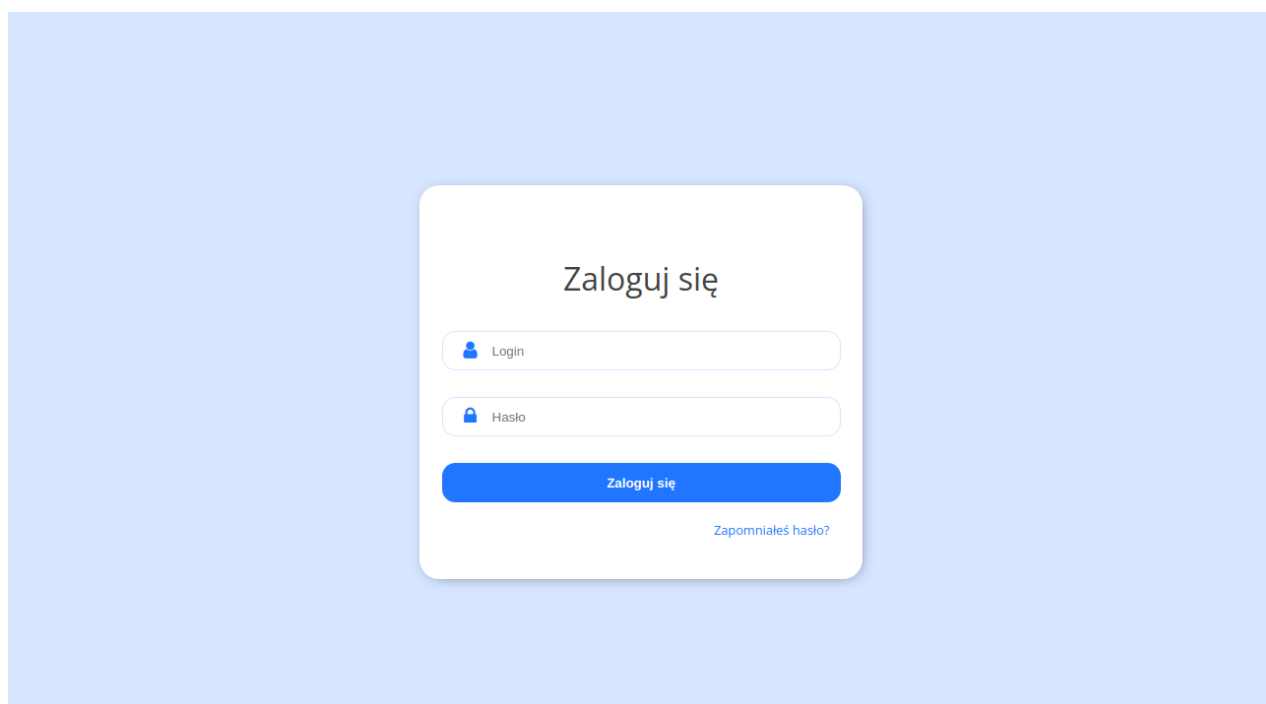
Biuro podróży

PRZYGOTOWANIE BAZY DAYCH

1. Utwórz bazę danych: `biuro_gr1` / `biuro_gr2` z ustawionym kodowaniem `utf8_unicode_ci`.
2. Do utworzonej bazy zaimportuj plik: `biuro.sql`.
3. Wykonaj zrzut ekranu z widocznym efektem importu. Zrzut zapisz pod nazwą `import.png`.
4. Utwórz użytkownika `pracownik_gr1@localhost` / `pracownik_gr2@localhost` z hasłem: `zaq1@WSX`. Wykonaj zrzut ekranu i zapisz pod nazwą `uzytkownik.png`.
5. Nadaj utworzonemu użytkownikowi uprawnienia modyfikacji, przeglądania, dodawania i usuwania danych do wszystkich tabel w bazie `biuro_gr1` / `biuro_gr2`.

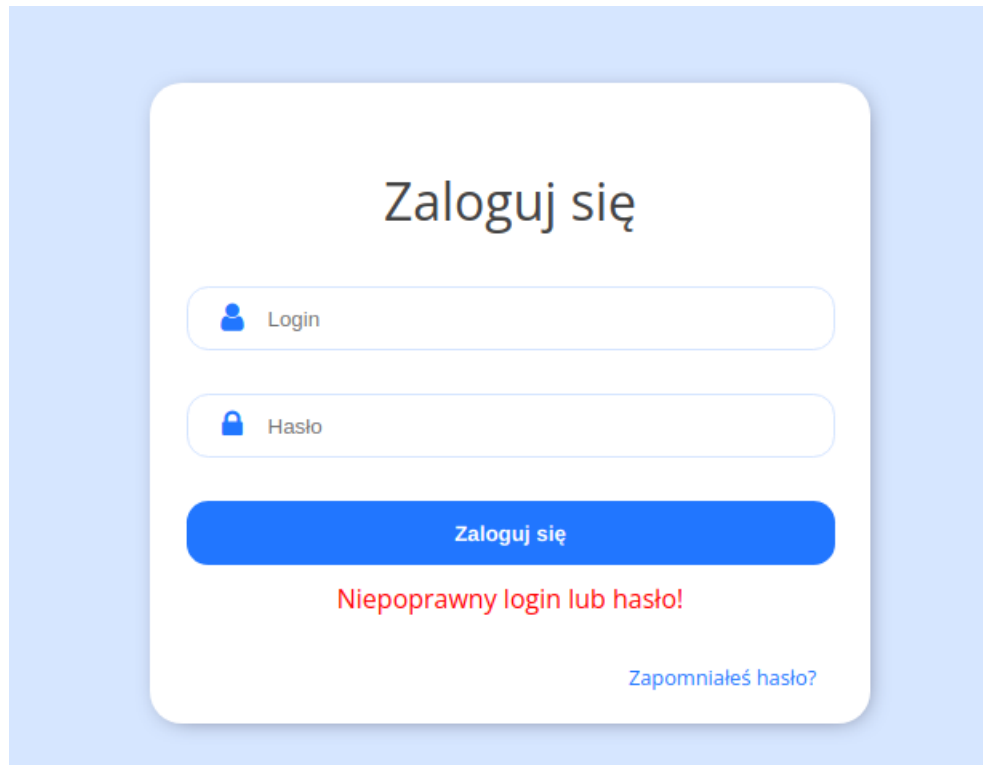
FORMULARZ LOGOWANIA

6. Utwórz plik `index.php`, ustaw poprawne kodowanie polskich znaków.
7. W pliku `index.php` utwórz formularz składający się z dwóch pól: pierwsze typu `text`, drugie typu `password` (oba wymagane) oraz przycisku z napisem: „Zaloguj się”.
8. Przycisk powinien wysyłać formularz do pliku `login.php` metodą `POST`.
9. Wygląd formularza podobny do formularza ze zdjęcia nr 1.



Zdjęcie nr 1 – Formularz logowania.

10. W prawym dolnym rogu, pod przyciskiem zaloguj znajduje się link prowadzący do pliku [reset_password.php](#). Link otwiera się w tej samej karcie.
11. Pomiędzy przyciskiem a linkiem do odzyskiwania hasła znajduje się paragraf, w którym w przypadku błędu wyświetlany jest komunikat. Informacja o wystąpieniu błędu odczytywana jest ze zmiennej sesyjnej o nazwie `$_SESSION['login_error']`.



Zdjęcie nr 2 – Komunikat o nieprawidłowych danych logowania.

SKRYPT LOGOWANIA

12. Skrypt PHP umieszczony w pliku [login.php](#).
13. Na początku działania, skrypt sprawdza czy użytkownik jest zalogowany. Sprawdzenie polega na weryfikacji czy istnieje ustawione `id` użytkownika w zmiennej `$_SESSION['logged_user']`.
 - a. Gdy zmienna jest ustawiona, skrypt przekierowuje użytkownika do strony [dashboard.php](#).
 - b. W przypadku nieudanej weryfikacji kontynuuje działanie.

Powyższy mechanizm sprawdzenia należy zastosować na każdej stronie, do której dostęp powinien mieć tylko zalogowany użytkownik!

W przypadku **a)** pozwalamy użytkownikowi wejść na stronę,
w przypadku **b)** należy przekierować użytkownika do strony logowania.

14. Skrypt łączy się z bazą danych utworzoną w poleceniu 1 za pomocą użytkownika utworzonego w poleceniu 5.
15. Skrypt sprawdza poprawność przesłanych z formularza danych (czy istnieją).
- Po prawidłowej weryfikacji skrypt przygotowuje a następnie wykonuje zapytanie do bazy danych, które zwraca id, login i hasło użytkownika o podanym w formularzu loginie.
 - W przypadku nieudanej weryfikacji skrypt ustawia w zmiennej sesyjnej `$_SESSION['login_error']` komunikat o błędzie i dokonuje przekierowania do pliku `index.php`
16. Skrypt przy użyciu metody `password_verify()` sprawdza poprawność podanego przez użytkownika hasła z zaszyfrowanym za pomocą `bcrypt` hasłem z bazy danych.
- Po prawidłowej weryfikacji skrypt:
 - W zmiennej sesyjnej `$_SESSION['logged_user']` zapisuje id zalogowanego użytkownika.
 - Przekierowuje użytkownika do strony `dashboard.php`.
 - W przypadku nieudanej weryfikacji skrypt ustawia w zmiennej sesyjnej `$_SESSION['login_error']` komunikat o błędzie i dokonuje przekierowania do pliku `index.php`
17. Na stronie `dashboard.php` znajduje się link, który prowadzi do pliku `logout.php`
18. W pliku `logout.php` skrypt niszczy ustawioną zmienną sesyjną `$_SESSION['logged_user']` oraz przekierowuje użytkownika do strony `index.php`

DASHBOARD

19. Na stronie `dashboard.php` skrypty pobiera z bazy danych login i funds z tabeli users, które wyświetla na stronie zgodnie z zdjęciem 3.
20. Na stronie `dashboard.php` skrypt sprawdza czy zalogowany użytkownik jest administratorem czy zwykłym użytkownikiem.
- Jeżeli użytkownik jest **administratorem** (isAdmin ma wartość 1), skrypt wykonuje:
 - Zapytanie do bazy danych, pobierające id oraz login użytkownika z tabeli users oraz id, offer_id, date z tabeli transactions.
 - Generuje tabelę wypełnioną pobranymi z bazy danymi, zgodnie z schematem: Zdjęcie 3
 - Jeżeli jest zwykłym użytkownikiem (isAdmin ma wartość 0), skrypt wykonuje:
 - Zapytanie do bazy danych, pobierające wszystkie dane z tabeli offer których cena jest mniejsza lub równa stanowi konta zalogowanego użytkownika (kolumna funds w tabeli users).
 - Generuje listę formularzy wypełnionych danymi zgodnie z zdjęciem 4.

ID	ID Użytkownika	Login	ID OFERTY	DATA
1	5	Anna	3	2023-01-01 12:14:02
2	2	Andrzej	1	2023-01-08 19:55:52
3	6	Qwert	1	2023-01-15 23:44:54

Zdjęcie 3 – zawartość strony dashborad.php dla kont administratorów

ID	NAZWA	MIEJSCE DOCELOWE	CENA	AKCJE
1	AlbatrosPalace	Egipt	2500.0	KUP TERAZ
3	Oz Side Premium	Turcja	7500.0	KUP TERAZ
7	Mitsis Norida Beach	Grecja	5600.0	KUP TERAZ

Zdjęcie 4 – zawartość strony dashboard.php dla zwykłego użytkownika

21. Widoczna na zdjęciu 4 lista formularzy składa się z inputów z atrybutem tylko do odczytu. Każdy wiersz jest osobnym formularzem, którego wysłanie prowadzi do pliku [order.php](#). Przycisk KUP TERAZ to input typu submit, który wysyła formularz metodą POST do wspomnianego wyżej pliku.

OBSŁUGA ZAMÓWIEŃ

22. W pliku [order.php](#) skrypt PHP odbiera id zamawianej wycieczki i umieszcza w bazie danych w tabeli [transaction](#) dane dotyczące tego zamówienia. (id użytkownika mamy w zmiennej sesyjnej, id oferty otrzymujemy z formularza, datę i id uzupełnia baza – wpisujemy wartość NULL). Po dokonaniu wpisu w bazie danych wyświetlamy użytkownikowi komunikat widoczny na zdjęciu nr 5, wraz z linkiem prowadzącym do strony [dashboard.php](#).

23. Po złożeniu zamówienia skrypt aktualizuje stan konta użytkownika obniżając go o cenę zakupionej wycieczki.

Wycieczka została kupiona!

[\[Powrót\]](#)

Zdjęcie 5 – komunikat na stronie order.php po złożeniu zamówienia.