

# Rejestracja i logowanie

Czas: 2 × 45 min

praca indywidualna

**Cel:** Zbudowanie kompletnego mechanizmu kont użytkowników w PHP: rejestracja z bezpiecznym przechowywaniem haseł, logowanie z sesją, ochrona podstron i podział na role.

**Po zajęciach potrafisz:**

- zapisuje hasła wyłącznie jako skróty `password_hash()` i sprawdza je `password_verify()`
- waliduje dane rejestracji po stronie serwera i wykrywa zajęty login lub e-mail
- loguje użytkownika do sesji, zmieniając jej identyfikator `session_regenerate_id(true)`
- chroni podstrony funkcjami wymagającymi zalogowania lub roli admin (kod 403)
- poprawnie wylogowuje użytkownika, usuwając dane sesji i ciasteczko

## Sprzęt i materiały

- komputer z Windows 11, XAMPP (Apache + MariaDB), przeglądarka z narzędziami F12
- plik `r2_12_portal.sql` i folder `r2_12_start` z folderu Pliki\_do\_zadan

## Oprogramowanie

- **XAMPP** — Apache, PHP 8.2, MariaDB, phpMyAdmin  
[apachefriends.org/download.html](https://apachefriends.org/download.html)
- **Visual Studio Code** — edycja plików PHP  
[code.visualstudio.com](https://code.visualstudio.com)

## Przygotowanie przed zajęciami

- Przypomnij sobie z wykładu 2.09 działanie sesji i funkcji `password_hash()` / `password_verify()` oraz wzorzec PRG z zadania 7.

## Bezpieczeństwo

- Do testów używaj wymyślonych kont i haseł — nigdy swoich prawdziwych haseł z innych serwisów.
- Nie zapisuj haseł jawnym tekstem ani w bazie, ani w plikach, ani w sprawozdaniu (zrzut tabeli pokazuje tylko skróty).
- Serwer XAMPP służy tylko do pracy lokalnej — nie udostępniaj go w sieci.

## Przebieg zadania

### 1 Przygotowanie projektu i bazy

Skopiuj folder `r2_12_start` do `C:\xampp\htdocs\portal`. Strony otwieraj przez `http://localhost/portal/`.

1. Zaimportuj `r2_12_portal.sql` w phpMyAdmin. Obejrzyj strukturę tabeli `uzytkownicy`: dlaczego kolumna `haslo` ma 255 znaków, a `login` i `email` mają indeks UNIQUE?
2. W `init.php` przed `session_start()` ustaw parametry ciasteczka sesji funkcją `session_set_cookie_params()` z tablicą: `lifetime 0`, `path '/'`, `httponly true`, `samesite 'Lax'`, `secure` zależne od tego, czy połączenie jest szyfrowane (`!empty($_SERVER['HTTPS'])`).

## 2 Rejestracja

Uzupełnij `rejestracja.php`. Komunikaty błędów zbieraj w tablicy `$bledy` — szablon już je wyświetla.

1. Walidacja: login pasuje do wyrażenia `/^[a-z0-9]{3,20}$/`, e-mail przechodzi `filter_var(..., FILTER_VALIDATE_EMAIL)`, hasło ma co najmniej 8 znaków (`mb_strlen`), oba hasła są identyczne. Po błędzie formularz zachowuje login i e-mail, ale nie hasła.
2. Zapytaniem przygotowanym sprawdź, czy login lub e-mail nie są już zajęte — pokaż osobny komunikat dla każdego przypadku.
3. Gdy dane są poprawne, zapisz użytkownika z hasłem `password_hash($haslo, PASSWORD_DEFAULT)`, ustaw komunikat `flash('Konto założone...')` i przekieruj do `logowanie.php` (PRG). Załóż dwa konta testowe i obejrzyj kolumnę `haslo` w phpMyAdmin — co oznacza początek `$2y$10$`?

## 3 Logowanie i strefa dla zalogowanych

Dane zalogowanego trzymaj w sesji: `$_SESSION['uid']`, `['login']`, `['rola']`.

```
function zalogowany(): bool { ... }  
function wymagaj_logowania(): void { ... } // brak sesji → flash + Location: logowanie.php  
function wymagaj_rola(string $rola): void { ... } // inna rola → kod 403
```

1. W `logowanie.php` pobierz użytkownika po loginie i sprawdź hasło `password_verify()`. Przy błędzie pokaż jeden wspólny komunikat „Nieprawidłowy login lub hasło.” bez względu na to, czy pomyłono login, czy hasło.
2. Po poprawnym logowaniu wywołaj `session_regenerate_id(true)`, zapisz dane w sesji, uaktualnij kolumnę `ostatnie_logowanie` i przekieruj do `panel.php`. W F12 → Application (Aplikacja) → Cookies zanotuj wartość `PHPSESSID` przed i po zalogowaniu.
3. Dopisz w `init.php` trzy funkcje z ramki powyżej. Utwórz `panel.php` (dane zalogowanego z bazy w tabeli) chroniony przez `wymagaj_logowania()`. Sprawdź wejście na `panel.php` bez logowania.

## 4 Role, menu i wylogowanie

1. Utwórz `admin.php` z listą wszystkich użytkowników (bez kolumny hasła), chroniony przez `wymagaj_rola('admin')`. Wejdź na nią jako zwykły użytkownik — w F12 → Network sprawdź kod odpowiedzi.
2. Nadaj jednemu kontu rolę admin w phpMyAdmin (`UPDATE uzytkownicy SET rola = 'admin' WHERE login = '...';`). Czy zmiana działa od razu, czy dopiero po ponownym zalogowaniu? Dlaczego?
3. Zmień `menu.php`: gość widzi Rejestracja i Logowanie; zalogowany — Panel, swój login i przycisk Wyloguj; admin dodatkowo odnośnik Administracja.
4. Utwórz `wyloguj.php` obsługujący tylko POST (formularz z przyciskiem w menu): wyczyść `$_SESSION`, usuń ciasteczko sesji (`setcookie` z datą w przeszłości), wywołaj `session_destroy()` i przekieruj na stronę startową.
5. Wykonaj zrzuty: błędy rejestracji, tabela `uzytkownicy` w phpMyAdmin, panel, strona admina, odpowiedź 403 w F12. Spakuj folder `portal` jako `nazwisko_imie_zad12.zip`.

## W sprawozdaniu umieść

- ☐ tabela wykonanych poleceń (numer kroku i punktu, np. 2.3) z kluczowymi fragmentami kodu
- ☐ zrzuty: błędy walidacji, tabela `uzytkownicy` ze skrótami haseł, panel, strona admina, kod 403 w F12
- ☐ wartości `PHPSESSID` przed i po zalogowaniu z komentarzem
- ☐ archiwum `nazwisko_imie_zad12.zip` z projektem
- ☐ odpowiedzi na pytania kontrolne

## Pytania kontrolne

1. Dlaczego hasła nie szyfruje się ani nie zapisuje skrótem MD5, tylko używa `password_hash()`? Co zawiera zapisany skrót oprócz samego wyniku?
2. Po co zmieniać identyfikator sesji tuż po zalogowaniu?
3. Dlaczego przy nieudanym logowaniu podajemy jeden wspólny komunikat zamiast „nie ma takiego loginu”?
4. Czym różni się ukrycie odnośnika „Administracja” w menu od sprawdzenia roli w `admin.php`? Które jest zabezpieczeniem?

## Kryteria oceny

| Kryterium                                                                                                       | Punkty    |
|-----------------------------------------------------------------------------------------------------------------|-----------|
| Rejestracja: walidacja 4 warunków, zachowanie pól, wykrycie zajętego loginu i e-maila                           | 5         |
| Hasła zapisane <code>password_hash()</code> , logowanie z <code>password_verify()</code> i wspólnym komunikatem | 4         |
| Sesja: parametry ciasteczka, <code>session_regenerate_id(true)</code> , panel chroniony przed gościem           | 4         |
| Rola admin: strona admina z kodem 403 dla innych, menu zależne od stanu i roli                                  | 3         |
| Wylogowanie POST usuwające sesję i ciasteczko; zrzuty i archiwum                                                | 2         |
| Odpowiedzi na pytania kontrolne                                                                                 | 2         |
| <b>Razem</b>                                                                                                    | <b>20</b> |

Sprawozdanie przygotuj na wzorze dołączonym do zadania (plik .docx o tym samym numerze).