

Portal pracy w Niemczech

Błażej Koceniak & Mateusz Ginter

1.Wstęp

1.1. Cel projektu

Głównym celem projektu jest stworzenie portalu internetowego umożliwiającego użytkownikom wyszukiwanie ofert pracy w Niemczech, a pracodawcom – publikowanie ogłoszeń rekrutacyjnych. Portal ma ułatwić proces poszukiwania zatrudnienia za granicą poprzez:

Dostęp do aktualnych ofert pracy w różnych branżach i regionach Niemiec.

Filtrowanie ogłoszeń według kryteriów (np. stanowisko, lokalizacja, wynagrodzenie).

Profil użytkownika (CV, doświadczenie, umiejętności).

System aplikowania na oferty bezpośrednio przez portal.

Narzędzia dla pracodawców (dodawanie ofert, zarządzanie kandydatami).

1.2. Zakres projektu Projekt obejmuje:

Stronę główną z wyszukiwarką ofert i najnowszymi ogłoszeniami.

Panel użytkownika (rejestracja, logowanie, zarządzanie profilem).

Panel pracodawcy (dodawanie i edycja ofert, przegląd aplikacji).

Moduł wyszukiwania z zaawansowanymi filtrami.

System powiadomień (np. o nowych ofertach pasujących do profilu).

Responsywny design (dostosowanie do urządzeń mobilnych).

1.3. Użytkownicy systemu

Portal skierowany jest do trzech głównych grup użytkowników:

Kandydaci – osoby poszukujące pracy w Niemczech:

Mogą przeglądać oferty, aplikować na stanowiska i zarządzać swoim profilem.

Pracodawcy – firmy i agencje rekrutacyjne:

Mogą publikować oferty pracy, przeglądać CV i kontaktować się z kandydatami.

Goście – osoby niezalogowane:

Mogą przeglądać podstawowe oferty, ale do aplikowania wymagana jest rejestracja.

2.Opis ogólny

2.1. Przeznaczenie witryny

Portal pracy w Niemczech ma na celu ułatwienie:

Kandydatom – znalezienie pracy za granicą poprzez dostęp do aktualnych ofert i narzędzi do aplikowania.

Pracodawcom – efektywną rekrutację pracowników z Polski dzięki łatwemu zarządzaniu ogłoszeniami i kandydatami.

Agencjom rekrutacyjnym – promocję ofert i kontakt z potencjalnymi pracownikami

2.2. Główne funkcjonalności

Dla kandydatów: Rejestracja i logowanie (przez e-mail lub konta społecznościowe).

Profil użytkownika z możliwością dodania CV, opisu doświadczenia i umiejętności.

Wyszukiwarka ofert z filtrami (branża, lokalizacja, typ umowy, wynagrodzenie).

System aplikowania na oferty (wysyłanie CV i listu motywacyjnego).

Dla pracodawców:

Panel rekrutera do dodawania i edycji ogłoszeń.

Przegląd zgłoszeń kandydatów wraz z ich profilami.

Narzędzia komunikacji (czat, e-maile).

Dla gości (niezalogowanych):

Przeglądanie podstawowych ofert pracy.

Dostęp do artykułów i poradników.

Możliwość założenia konta.

2.3. Wymagania techniczne

Dodatkowe technologie:

Frontend: HTML, CSS, JavaScript .

Backend: PHP lub Node.js.

3.Struktura strony internetowej

3.1. Strona główna (index.php) Funkcjonalności: Wyświetlanie losowych ofert pracy (2 rekordy z bazy danych)

Pokazywanie najnowszych opinii użytkowników (3 ostatnie opinie)

Wyszukiwarka ofert z filtrami:

Stanowisko/firma

Lokalizacja (miasto/region)

Slider opinii z automatycznym przewijaniem co 5 sekund

Struktura pliku: Nagłówek (Header):

Logo portalu

Menu nawigacyjne (dostosowane do roli użytkownika:
admin/pracodawca/zwykły użytkownik)

Sekcja wyszukiwania:

html Szukaj Sekcja popularnych ofert:

Wyświetla 2 losowe oferty z bazy danych (SELECT ... ORDER BY RAND() LIMIT 2)

Sekcja opinii:

Slider z 3 najnowszymi opiniami (JOIN z tabelą użytkowników)

Stopka:

Informacje o prawach autorskich

Linki do regulaminu i polityki prywatności

Technologie: PHP (pobieranie danych z MySQL)

JavaScript (obsługa wyszukiwarki i slidera opinii)

HTML5/CSS3 (responsywny layout)

3.2. Podstrona "Lista ofert" (lista_ofert.php) (przykład kontynuacji) Funkcjonalności:

Wyświetlanie wszystkich ofert pracy

Filtrowanie wyników (np. przez
?keyword=programista&location=Berlin)

Paginacja (stronicowanie wyników)

3.3. Podstrona "Szczegóły oferty" (oferta_szczegoly.php)
Funkcjonalności:

Pełny opis oferty (pobierany przez id z URL)

Przycisk "Aplikuj" (dla zalogowanych użytkowników)

3.4. Podstrona "Panel pracodawcy" (panel_pracodawcy.php)
Funkcjonalności:

Dodawanie/edycja ofert

Przegląd zgłoszeń kandydatów

3.5. Panel logowania i rejestracji (logowanie.php, rejestracja.php, admin.php) Logowanie:

Formularz z walidacją (e-mail/hasło)

Przekierowanie do odpowiedniego panelu
(admin/pracodawca/użytkownik)

Rejestracja:

Wybór typu konta (kandydat/pracodawca)

Weryfikacja e-maila

Panel admina (admin.php):

Zarządzanie użytkownikami i ofertami

Moderowanie opinii

Baza danych: MySQL (relacyjna).

Bezpieczeństwo: HTTPS.

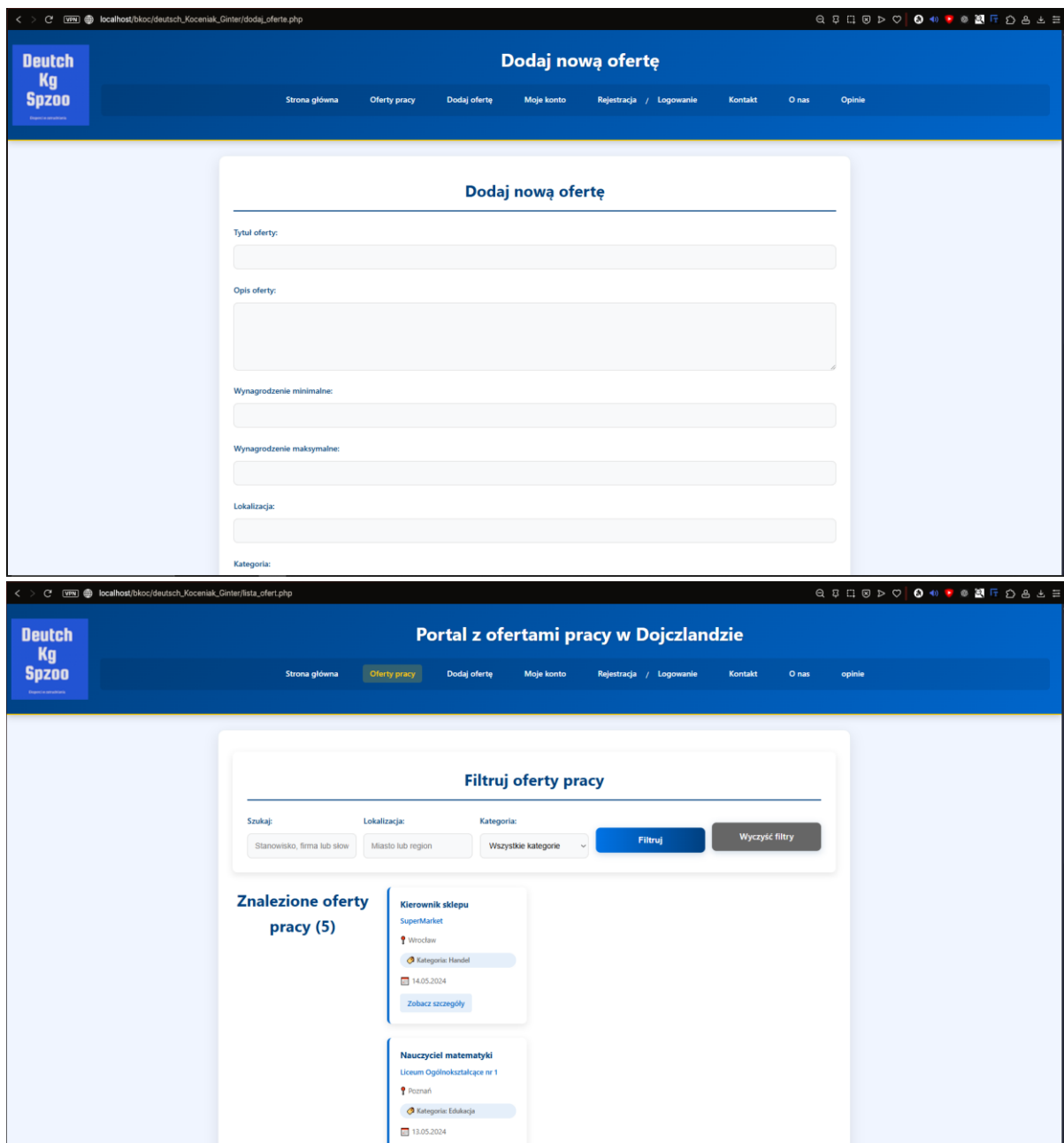
4.Elementy statyczne

4.2. CSS – stylizacja (style.css)

```
<!DOCTYPE html>
<html lang="pl">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Lista ofert pracy [ ] Dojczland</title>
  <link rel="stylesheet" href="styleindex.css">
  <link href="favicon.ico" rel="icon" type="image/x-icon">
  <style>
    /* Przyjemny, przyjemny dla zmyslow font */
```

```
<!DOCTYPE html>
<html lang="pl">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Dodaj ofertę</title>
  <link rel="stylesheet" href="styleindex.css">
  <link href="favicon.ico" rel="icon" type="image/x-icon">
</head>
<body>
```

```
<script>
```

5.JavaScript

Komentarze w pliku skrypty.js

6. Elementy dynamiczne (PHP)

6.1. Obsługa logowania i rejestracji użytkownika

logowanie

```
<?php
session_start();
require_once 'config.php';

if (isset($_SESSION['user_id'])) {
    header('Location: konto.php');
    exit();
}

$error = '';

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $email = trim($_POST['email']);
    $password = $_POST['password'];

    if (!empty($email) && !empty($password)) {
        $stmt = $conn->prepare("SELECT id, imie, nazwisko, email, haslo, rola FROM uzytkownicy WHERE email = ?");
        $stmt->bind_param('s', $email);
        $stmt->execute();
        $result = $stmt->get_result();

        if ($result->num_rows === 1) {
            $user = $result->fetch_assoc();
            if ($password === $user['haslo']) {
                $_SESSION['user_id'] = $user['id'];
                $_SESSION['email'] = $user['email'];
                $_SESSION['imie'] = $user['imie'];
                $_SESSION['nazwisko'] = $user['nazwisko'];
                $_SESSION['rola'] = $user['rola'];

                header('Location: ' . ($user['rola'] === 'admin' ? 'admin_panel.php' : 'konto.php'));
                exit();
            } else {
                $error = "Nieprawidłowy email lub hasło";
            }
        } else {
            $error = "Nieprawidłowy email lub hasło";
        }
        $stmt->close();
    } else {
        $error = "Proszę wypełnić wszystkie pola";
    }
}
?>
```

rejestracja

```

<?php
session_start();
require_once 'config.php';
if (isset($_SESSION['user_id'])) {
    header('Location: konto.php');
    exit();
}

$error = '';
$success = '';

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $imie = trim($_POST['imie']);
    $nazwisko = trim($_POST['nazwisko']);
    $email = trim($_POST['email']);
    $opis = trim($_POST['opis']);
    $password = $_POST['password'];
    $confirm = $_POST['confirm'];
    $rola = $_POST['rola'];
    $regulamin = isset($_POST['regulamin']) ? true : false;
    $telefon = trim($_POST['telefon']);

    if (empty($imie) || empty($nazwisko) || empty($email) || empty($password) || empty($confirm) || empty($opis) || empty($telefon)) {
        $error = 'Wszystkie pola są wymagane';
    } elseif (ipreg_match('/^[a-zA-Ząęłńóźżiäçłtńóśzłłs-]{2,50}$/i', $imie)) {
        $error = 'Imię może zawierać tylko litery (2-50 znaków)';
    } elseif (ipreg_match('/^[a-zA-Ząęłńóźżiäçłtńóśzłłs-]{2,50}$/i', $nazwisko)) {
        $error = 'Nazwisko może zawierać tylko litery i myślniki (2-50 znaków)';
    } elseif (ifilter_var($email, FILTER_VALIDATE_EMAIL)) {
        $error = 'Nieprawidłowy format adresu email';
    } elseif (ipreg_match('/^\d{7,15}$/i', $telefon)) {
        $error = 'Nieprawidłowy format numeru telefonu';
    } elseif ($password !== $confirm) {
        $error = 'Hasła nie są identyczne';
    } elseif (strlen($password) < 6) {
        $error = 'Hasło musi mieć co najmniej 6 znaków';
    } elseif (!$regulamin) {
        $error = 'Musisz zaakceptować regulamin';
    } else {
        try {
            $stmt = $conn->prepare("SELECT id FROM uzytkownicy WHERE email = ?");
            $stmt->bind_param('s', $email);
            $stmt->execute();
            $stmt->store_result();

            if ($stmt->num_rows > 0) {
                $error = 'Podany adres email jest już zarejestrowany';
            } else {
                $hashed_password = $password; // In a real system, use password_hash()

                $stmt = $conn->prepare("INSERT INTO uzytkownicy (imie, nazwisko, email, haslo, rola, opis, telefon) VALUES (?, ?, ?, ?, ?, ?, ?)");
                $stmt->bind_param('sssssss', $imie, $nazwisko, $email, $hashed_password, $rola, $opis, $telefon);

                if ($stmt->execute()) {
                    $success = 'Rejestracja zakończona pomyślnie. Możesz się teraz zalogować.';
                    $_POST = array(); // Clear form after successful registration
                } else {
                    $error = 'Wystąpił błąd podczas rejestracji: ' . $conn->error;
                }
            }
            $stmt->close();
        } catch (Exception $e) {
            $error = 'Wystąpił błąd: ' . $e->getMessage();
        }
    }
}
?>

```

6.2. Pobieranie danych z bazy (katalog.php)

```
// Pobierz oferty z bazy
$offers = [];
try {
    $stmt = $conn->prepare($query);

    if (!empty($params)) {
        $stmt->bind_param($types, ...$params);
    }

    $stmt->execute();
    $result = $stmt->get_result();

    while ($row = $result->fetch_assoc()) {
        $offers[] = $row;
    }

    $stmt->close();
} catch (Exception $e) {
    die("Wystąpił błąd: " . $e->getMessage());
}
```

6.3 Składanie aplikacji przez kandydata

Funkcjonalność

Strona aplikuj.php pozwala zalogowanym użytkownikom (kandydatom) złożyć aplikację na wybrane ogłoszenie o pracę.

Opis działania

Sprawdza, czy w URL przekazano id oferty. Jeśli nie, przekierowuje do listy ofert.

Pobiera tytuł oferty z bazy danych, aby wyświetlić go na stronie.

Obsługuje formularz metodą POST:

Jeśli użytkownik nie jest zalogowany, zapamiętuje adres powrotu i przekierowuje do logowania.

Pobiera wiadomość od kandydata oraz załącza pliki CV i opcjonalnie list motywacyjny.

Sprawdza rozszerzenia i rozmiar plików (do 2MB, formaty: pdf, doc, docx).

Zapisuje pliki w folderze uploads pod unikalnymi nazwami.

Jeśli wszystko jest poprawne, w jednej transakcji SQL:

Dodaje rekord aplikacji do tabeli aplikacje.

Pobiera ID pracodawcy odpowiedzialnego za ofertę.

Tworzy powiadomienie dla pracodawcy o nowej aplikacji.

Po pomyślnym zapisie przekierowuje użytkownika do strony potwierdzenia.

W razie błędów wyświetla je nad formularzem.

6.4 Panel administracyjny (dodawanie/edycja/usuwanie wpisów)

Panel administracyjny umożliwia administratorowi zarządzanie użytkownikami systemu. Dostęp do panelu mają tylko zalogowani użytkownicy z rolą admin.

Funkcjonalności:

Wyświetlenie listy wszystkich użytkowników z bazy danych w formie tabeli.

Edycja danych użytkownika: imię, nazwisko, email, rola (admin, użytkownik, pracodawca) oraz pensja.

Aktualizacja danych w bazie MySQL po zatwierdzeniu formularza edycji.

Ochrona panelu poprzez sprawdzenie sesji i roli użytkownika (przekierowanie do logowania, jeśli brak uprawnień).

Opis działania:

Po wejściu na stronę skrypt sprawdza, czy użytkownik jest zalogowany i czy ma rolę admin.

Jeśli warunki nie są spełnione, następuje przekierowanie na stronę logowania.

Dane użytkowników pobierane są z tabeli uzytkownicy w bazie danych i wyświetlane w tabeli HTML.

Każdy wiersz ma przycisk "Edytuj", który wywołuje funkcję JavaScript editUser(), wypełniającą formularz danymi użytkownika i wyświetlającą go.

Formularz pozwala zmienić dane i po zatwierdzeniu przesyła je metodą POST.

Skrypt PHP aktualizuje dane w bazie danych za pomocą zapytania SQL UPDATE.

Po aktualizacji użytkownik może ponownie edytować dane lub zamknąć formularz bez zmian.

6.4 Panel pracodawcy — przegląd aplikacji na ofertę pracy

Funkcjonalność

Strona umożliwia pracodawcy przeglądanie aplikacji kandydatów na wybraną ofertę pracy. Dostęp do niej mają tylko zalogowani użytkownicy z rolą pracodawca.

Opis działania

Po wejściu na stronę następuje weryfikacja, czy użytkownik jest zalogowany i ma odpowiednią rolę. Jeśli nie, jest przekierowywany na stronę główną.

Sprawdzone jest, czy w URL przekazano identyfikator oferty (id), jeśli nie, następuje przekierowanie do panelu pracodawcy.

Na podstawie przekazanego id pobierane są z bazy danych wszystkie aplikacje związane z daną ofertą, wraz z danymi użytkowników (imię, nazwisko).

Wyniki wyświetlane są w tabeli z kolumnami:

ID aplikacji

Imię i nazwisko kandydata

Wiadomość

Link do pobrania CV

Link do pobrania listu motywacyjnego

Data aplikacji

Status aplikacji (zmieniany na żywo poprzez rozwijane menu)

Zmiana statusu aplikacji jest obsługiwana przez formularz POST wysyłany do skryptu `update_status.php`.

7. Baza danych

7.1. Projekt struktury bazy danych

Baza danych portalu pracy w Niemczech składa się z 12 tabel, które umożliwiają zarządzanie:

Użytkownikami (kandydatami, pracodawcami, adminami).

Ofertami pracy i kategoriami.

Aplikacjami, opiniami, powiadomieniami i wiadomościami.

Umiejętnościami użytkowników i ulubionymi ofertami.

Kluczowe założenia: Relacyjność – połączenia między tabelami przez klucze obce (FOREIGN KEY).

Normalizacja – unikanie redundancji danych (np. oddzielne tabele dla kategorii i umiejętności).

Bezpieczeństwo – hasła użytkowników przechowywane jako hashe (w przykładowych danych użyto zwykłego tekstu dla testów).

7.2. Opis tabel i pól

Główne tabele: Tabela Opis Kluczowe pola użytkownicy Dane użytkowników (kandydaci, pracodawcy, admini). id, email, hasło, rola (uzytkownik/pracodawca/admin). oferty Oferty pracy dodawane przez pracodawców. id, tytuł, firma, wynagrodzenie_min/max, id_pracodawcy. aplikacje Aplikacje użytkowników na oferty. id_uzytkownika, id_oferty, status (np. zaakceptowana). kategorie Kategorie ofert (np. IT, Budownictwo). id, nazwa. opinie Opinie użytkowników o portalu. id_uzytkownika, ocena (1-5), tresc. wiadomosci Wiadomości między użytkownikami. id_nadawcy, id_odbiiorcy, temat, tresc. Pozostałe tabele: oferty_kategorie – łączy oferty z kategoriami (relacja wiele-do-wielu).

ulubione_oferty – ulubione oferty użytkowników.

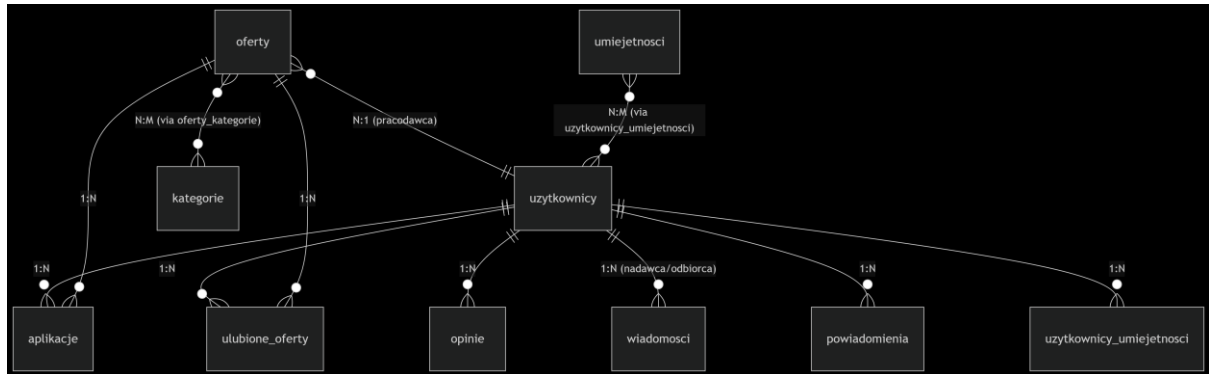
umiejtnosci – lista umiejętności (np. PHP, JavaScript).

uzytkownicy_umiejtnosci – powiązania użytkowników z umiejętnościami.

powiadomienia – powiadomienia systemowe (np. nowa oferta).

kontakt – wiadomości od gości (formularz kontaktowy).

7.3. Relacje między tabelami



Kluczowe zależności:

aplikacje:

id_uzytkownika → uzytkownicy(id)

id_oferty → oferty(id)

oferty:

id_pracodawcy → uzytkownicy(id)

oferty_kategorie:

id_oferty → oferty(id)

id_kategorii → kategorie(id)

ulubione_oferty:

id_uzytkownika → uzytkownicy(id)

id_oferty → oferty(id)

10.Załączniki

10.1. Fragmenty kodu źródłowego (HTML/JS/PHP/CSS) z komentarzami

```

<?php
session_start();
$user_id = $_SESSION['user_id'];
$user_data = [];

/**
 * Skrypt strony głównej portalu z ofertami pracy
 *
 * Funkcjonalności:
 * - Pobieranie losowych ofert pracy
 * - Pobieranie najnowszych opinii użytkowników
 * - Wyszukiwarka ofert
 * - Wyświetlanie slidera z opiniami
 */

// Dołączenie pliku konfiguracyjnego z danymi połączenia do bazy
require_once 'config.php';

/**
 * Pobieranie 2 losowych ofert pracy
 * Zapytanie wykorzystuje ORDER BY RAND() do losowego wyboru rekordów
 */
$popularne_oferty = [];
try {
    $stmt = $conn->prepare("SELECT o.id, o.tytul, o.firma, o.lokalizacja, o.opis
                            FROM oferty o
                            ORDER BY RAND() LIMIT 2");
    $stmt->execute();
    $result = $stmt->get_result();

    while ($row = $result->fetch_assoc()) {
        $popularne_oferty[] = $row;
    }
} catch (Exception $e) {
    die("Wystąpił błąd: " . $e->getMessage());
}

/**
 * Pobieranie 3 najnowszych opinii użytkowników
 * Opinie są łączone z tabelą użytkowników, aby pobrać dane autora
 */
$opinie = [];
try {
    $stmt = $conn->prepare("SELECT o.tresc, u.imie, u.nazwisko
                            FROM opinie o
                            JOIN uzytkownicy u ON o.id_uzytkownika = u.id
                            ORDER BY o.data_dodania DESC LIMIT 3");
    $stmt->execute();
    $result = $stmt->get_result();

    while ($row = $result->fetch_assoc()) {
        $opinie[] = $row;
    }
} catch (Exception $e) {
    die("Wystąpił błąd: " . $e->getMessage());
}
?>

```

```

/* Stylizacja głównego panelu konta - siatka 2 kolumny z odstępem między nimi */
.konto-panel {
  display: grid;
  grid-template-columns: 1fr 1fr; /* Dwie równe kolumny */
  gap: 30px; /* Odstęp między kolumnami */
  margin-top: 20px; /* Górny margines panelu */
}

/* Stylizacja sekcji danych osobowych i ustawień konta */
.dane-osobowe, .ustawienia-konta {
  background-color: #fafafa; /* Jasne tło */
  border: 1px solid #ddd; /* Delikatna ramka */
  border-radius: 5px; /* Zaokrąglone rogi */
  padding: 20px; /* Wewnętrzny odstęp */
  box-shadow: 0 2px 5px rgba(0,0,0,0.05); /* Subtelny cień */
}

/* Nagłówki sekcji danych i ustawień */
.dane-osobowe h3, .ustawienia-konta h3 {
  color: #004080; /* Kolor ciemnoniebieski */
  border-bottom: 2px solid #004080; /* Dolna linia pod nagłówkiem */
  padding-bottom: 8px; /* Odstęp pod tekstem */
  margin-bottom: 15px; /* Margines pod nagłówkiem */
}

/* Akapit w sekcji danych osobowych */
.dane-osobowe p {
  margin-bottom: 12px; /* Odstęp między paragrafami */
  line-height: 1.6; /* Wysokość linii dla lepszej czytelności */
}

/* Elementy wyróżnione (np. etykiety danych) */
.dane-osobowe strong {
  color: #004080; /* Kolor tekstu */
  min-width: 80px; /* Minimalna szerokość */
  display: inline-block; /* Pozwala ustawić szerokość i wyrównać tekst */
}

```

```

// Nowa wersja - maksymalnie kompatybilna
function initSlider() {
    const opinie = document.getElementsByClassName('opinia');
    let current = 0;

    // Pokazuje aktualną opinię
    function showOpinion(index) {
        for (let i = 0; i < opinie.length; i++) {
            opinie[i].classList.remove('aktywna');
            opinie[i].style.display = 'none';
        }
        opinie[index].classList.add('aktywna');
        opinie[index].style.display = 'block';
    }

    // Inicjalizacja
    showOpinion(0);

    // Event listeners dla przycisków
    document.getElementById('poprzednia').onclick = function() {
        current = (current - 1 + opinie.length) % opinie.length;
        showOpinion(current);
    };

    document.getElementById('nastepna').onclick = function() {
        current = (current + 1) % opinie.length;
        showOpinion(current);
    };
}

// Uruchom slider gdy strona się ładuje
if (document.readyState === 'loading') {
    document.addEventListener('DOMContentLoaded', initSlider);
} else {
    initSlider();
}

// Obsługa formularza (przykładowa)
document.querySelector('form')?.addEventListener('submit', function(e) {
    e.preventDefault();
    alert('Formularz został wysłany!');
    // Tutaj można dodać faktyczną logikę wysyłania formularza
});

```

```

<!DOCTYPE html>
<html lang="pl">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Szczegóły oferty - <?= htmlspecialchars($oferta['tytul']) ?></title>
    <!-- Główny arkusz stylów strony -->
    <link rel="stylesheet" href="styleindex.css">
    <link href="favicon.ico" rel="icon" type="image/x-icon">
</head>
<body>

<!-- Nagłówek strony z menu nawigacyjnym -->
<header>
    
    <h1>Portal z ofertami pracy w Dojczlandzie</h1>
    <nav>
        <ul>
            <!-- Podstawowe linki nawigacyjne -->
            <li><a href="index.php">Strona główna</a></li>
            <li><a href="lista_ofert.php">Oferty pracy</a></li>
            <li><a href="dodaj_oferte.php">Dodaj ofertę</a></li>
            <li><a href="konto.php">Moje konto</a></li>
            <li><a href="rejestracja.php">Rejestracja</a> / <a href="logowanie.php">Logowanie</a></li>
            <li><a href="kontakt.php">Kontakt</a></li>
            <li><a href="o_nas.php">O nas</a></li>
            <li><a href="opinie.php">opinie</a></li>
            <!-- Link do panelu admina widoczny tylko dla administratorów -->
            <?php if (isset($_SESSION['rola']) && $_SESSION['rola'] === 'admin'): ?>
            <li><a href="admin_panel.php">Panel Admina</a></li>
            <?php endif; ?>
            <?php if (isset($_SESSION['rola']) && $_SESSION['rola'] === 'pracodawca'): ?>
            <li><a href="panel_pracodawcy.php">panel pracodawcy</a></li>
            <?php endif; ?>
        </ul>
    </nav>
</header>

<!-- Główna zawartość strony -->
<main>
    <!-- Sekcja szczegółów oferty -->
    <section id="szczegoly-oferty">
        <h2><?= htmlspecialchars($oferta['tytul']) ?></h2>

        <div class="oferta-szczegoly">
            <!-- Nagłówek oferty z podstawowymi informacjami -->
            <div class="oferta-naglowek">
                <h3><?= htmlspecialchars($oferta['firma']) ?> • <?= htmlspecialchars($oferta['lokalizacja']) ?></h3>
                <p class="data-dodania">Dodano: <?= date('d.m.V', strtotime($oferta['data_dodania'])) ?></p>

                <!-- Wyświetlanie kategorii jeśli istnieją -->
                <?php if (!empty($kategorie)): ?>
                <div class="kategorie">
                    <span>Kategorie: </span>
                    <?php foreach ($kategorie as $kategoria): ?>
                    <span class="kategoria"><?= htmlspecialchars($kategoria) ?></span>
                    <?php endforeach; ?>
                </div>
                <?php endif; ?>
            </div>
        </div>
    </section>

```