

Strona tytułowa

Spis treści

MeatMasters

Autorzy: Dominik Bogdan, Kacper Bury

Spis treści

1. [Wstęp](#)
2. [Opis ogólny](#)
3. [Struktura strony internetowej](#)
4. [Elementy statyczne](#)
5. [JavaScript](#)
6. [Elementy dynamiczne \(PHP\)](#)
7. [Baza danych](#)
8. [Bezpieczeństwo](#)
9. [Możliwości rozbudowy](#)
10. [Załączniki](#)

1. Wstęp

1.1 Cel projektu

Celem projektu "MeatMasters" jest opracowanie kompleksowego systemu internetowego dla hurtowni mięsa. System ten ma umożliwiać użytkownikom przeglądanie oferty produktów, kontakt z firmą, zamawianie produktów, a także logowanie się do swojego konta klienta. Dla administratorów i pracowników system stanowi narzędzie zarządzania treścią oraz klientami. Dla kierowników także system zarządzania pracownikami.

Projekt obejmuje zaprojektowanie i wdrożenie kompletnej strony internetowej obsługiwanej przez bazę danych. Zrealizowane zostały następujące funkcjonalności:

- przeglądanie oferty produktowej (strona główna i katalog),
- rejestracja i logowanie klientów,
- bezpieczne logowanie pracowników (oddzielna tabela),
- panel użytkownika (klienta) z możliwością edycji danych i wylogowania,
- panel pracownika z możliwością dodawania nowych pracowników,
- możliwość dodawania opinii o firmie przez klienta,
- formularz kontaktowy z automatycznym wypełnianiem danych z bazy,
- obsługa sesji użytkownika i przypisywanie ról,
- możliwość dynamicznego ładowania danych z bazy (np. lista produktów, opinie).

System został wykonany przy użyciu HTML, CSS, JavaScript oraz PHP i działa lokalnie w środowisku XAMPP z bazą MySQL.

1.3 Użytkownicy systemu W systemie „MeatMasters” wyróżniamy dwie podstawowe grupy użytkowników:

- Klient
 - Ma możliwość rejestracji oraz logowania,
 - Po zalogowaniu uzyskuje dostęp do własnego panelu użytkownika (profil.php),
 - Może przysłać zapytania przez formularz kontaktowy (kontakt.php),
 - Może dodać swoją opinię o firmie (opinie.php).

- **Pracownik**

- Loguje się przez formularz logowania (logowanie.php),
- Po uwierzytelnieniu otrzymuje dostęp do panelu pracowniczego (pracownicy_p.php),
- Może przeglądać wiadomości od klientów oraz zarządzać podstawowymi funkcjami systemu (np. przegląd danych kontaktowych).

Użytkownicy z uprawnieniami kierownika mają dodatkowe możliwości, takie jak dodawanie nowych pracowników.

Rola użytkownika jest ustalana na podstawie danych logowania i zapisywana w sesji po pomyślnym zalogowaniu.

2. Opis ogólny

2.1 Przeznaczenie witryny Strona internetowa służy jako platforma informacyjna i komunikacyjna hurtowni mięsnej MeatMasters. Pozwala klientom zapoznać się z ofertą firmy, składać zamówienia, a także umożliwia kontakt i pozostawienie opinii. Pracownicy mają dostęp do narzędzi administracyjnych.













































2.2 Główne funkcjonalności

- Wyświetlanie oferty produktowej,
- Aktualnościami,
- Formularz kontaktowy,
- Możliwość rejestracji i logowania klientów,
- Panel użytkownika,
- Panel pracownika,
- Opinie klientów,
- Składanie zamówień przez klientów.

2.3 Wymagania techniczne

- XAMPP,
- PHP 7.0+, MySQL,
- Przeglądarka internetowa (Chrome, Firefox, Edge),
- Edytor kodu (Visual Studio Code)

2.4

 .gitignore	25.05.2025 02:48	Dokument tekstowy	1 KB
 aktualnosci	25.05.2025 02:48	PHP Source File	9 KB
 db	25.05.2025 02:48	PHP Source File	1 KB
 drób	25.05.2025 02:48	Plik JPG	169 KB
 faq	25.05.2025 02:48	Plik JPG	9 236 KB
 faq	25.05.2025 02:48	PHP Source File	14 KB
 filet-z-kurczaka	25.05.2025 15:08	Plik JPG	14 KB
 firma	25.05.2025 02:48	Plik PNG	1 600 KB
 icon	25.05.2025 02:48	Plik PNG	18 KB
 kabab-drobiowy	25.05.2025 15:08	Plik JPG	56 KB
 karkowka	25.05.2025 15:08	Plik JPG	25 KB
 kebabwol	25.05.2025 15:08	Plik JPG	5 KB
 kontakt	25.05.2025 02:48	PHP Source File	15 KB
 kontakty_p	25.05.2025 02:48	PHP Source File	16 KB
 koszyk	25.05.2025 02:48	PHP Source File	21 KB
 Logo	25.05.2025 15:08	Plik PNG	50 KB
 logowanie	25.05.2025 02:48	PHP Source File	13 KB
 meatmasters	25.05.2025 02:48	SQL Text File	28 KB
 miecho_tlo	25.05.2025 15:08	Plik JPG	278 KB
 miecho1	25.05.2025 15:08	Plik AVIF	94 KB
 mielonka-wolowa	25.05.2025 15:08	Plik JPG	270 KB
 o_nas	25.05.2025 02:48	PHP Source File	8 KB
 Oferta	25.05.2025 02:48	PHP Source File	13 KB
 opinie	25.05.2025 02:48	PHP Source File	15 KB
 plan	25.05.2025 02:48	Dokument Adobe ...	432 KB
 pracownicy_p	25.05.2025 02:48	PHP Source File	28 KB
 profil	25.05.2025 02:48	PHP Source File	20 KB
 README	25.05.2025 02:48	Markdown Source...	1 KB
 rejestrowanie	25.05.2025 02:48	PHP Source File	15 KB
 reklamacje_p	25.05.2025 02:48	PHP Source File	14 KB
 rostbef	25.05.2025 15:08	Plik JPG	67 KB
 schab	25.05.2025 15:08	Plik JPG	55 KB
 sesje	25.05.2025 02:48	PHP Source File	3 KB
 sklep	25.05.2025 02:48	Plik JavaScript	4 KB
 sklep	25.05.2025 02:48	PHP Source File	23 KB
 # sklep_style	25.05.2025 15:08	CSS Source File	8 KB
 Strona_glowna	25.05.2025 02:48	PHP Source File	8 KB
 # style	25.05.2025 02:48	CSS Source File	8 KB
<hr/>			
 # style_oferta	25.05.2025 15:08	CSS Source File	7 KB
 # style-o_nas	25.05.2025 02:48	CSS Source File	3 KB
 towar_p	25.05.2025 02:48	PHP Source File	14 KB
 udka-z-kurczaka	25.05.2025 15:08	Plik JPG	6 KB
 zamowienia_p	25.05.2025 02:48	PHP Source File	11 KB
 zamowienie	25.05.2025 02:48	PHP Source File	6 KB

- Strona_glowna.php – Strona główna serwisu.
 - logowanie.php – Logowanie użytkowników (klienci, pracownicy).
 - rejestrowanie.php – Rejestracja nowych klientów.
 - profil.php – Panel użytkownika-klienta.
 - pracownicy_p.php – Panel pracownika.
 - kontakt.php – Formularz kontaktowy.
 - kontakty_p.php – Panel do obsługi kontaktów.
 - sklep.php – Strona sklepu z produktami.
 - koszyk.php – Koszyk zakupowy klienta.
 - zamowienie.php – Obsługa składania zamówienia.
 - zamowienia_p.php – Lista zamówień dla pracownika.
 - reklamacje_p.php – Obsługa reklamacji (panel pracownika).
 - opinie.php – Dodawanie opinii przez klienta.
 - faq.php – Strona z pytaniami i odpowiedziami.
 - o_nas.php – Informacje o firmie.
 - towar_p.php – Obsługa towaru (zarządzanie zawartością).
 - aktualnosci.php – Sekcja aktualności i komunikatów.
 - sesje.php – Obsługa sesji użytkowników.
-
- db.php – Plik do połączenia z bazą danych.
 - drób.jpg, filet-z-kurczaka.jpg, kabab-drobiowy.jpg, karkowka.jpg, mielonka-wolowa.jpg, rostbef.jpg, schab.jpg, uдка-z-kurczaka.jpg – Zdjęcia produktów.
 - firma.png, icon.png, Logo.png – Ikony i logo firmy.

- miecho1.avif, miecho_tlo.jpg – Tła i grafiki dekoracyjne.
- faq.jpg – Ilustracja do sekcji FAQ.
- meatmasters.sql – Struktura i dane startowe bazy (klienci, pracownicy, zamówienia, itp.).
- style.css – Główny arkusz stylów dla całego serwisu.
- style_oferta.css – Stylizacja strony z ofertą produktów.
- style-o_nas.css – Stylizacja sekcji „O nas”.
- sklep_style.css – Stylizacja sklepu internetowego.
- sklep.js – Skrypty odpowiedzialne za działanie dynamicznych elementów w sklepie.

3.1 Strona główna (strona_glowna.php)

- Zawiera informacje o firmie, aktualności, skróty do sklepu i kontaktu.
- Pokazuje losowe lub promowane produkty.
- Posiada nagłówek z nawigacją, logo oraz stopkę z danymi kontaktowymi.

3.2 Podstrona: Sklep (sklep.php)

- Wyświetla wszystkie dostępne produkty z bazy danych.
- Produkty można filtrować według kategorii (np. drób, wieprzowina).
- Użytkownik może dodać produkty do koszyka.

3.3 Podstrona: Koszyk (koszyk.php)

- Pokazuje produkty dodane przez użytkownika.
- Umożliwia edycję ilości, usuwanie pozycji oraz przejście do zamówienia.
- Wymaga zalogowania, by złożyć zamówienie.

3.4 Podstrona: Kontakt i opinie (*kontakt.php, opinie.php*)

- kontakt.php – formularz kontaktowy z automatycznym uzupełnianiem danych (dla zalogowanych).
- opinie.php – użytkownik może dodać opinię o firmie, która trafia do bazy danych i jest widoczna na stronie.

3.5 Panel logowania i rejestracji (*logowanie.php, rejestrowanie.php, pracownicy_p.php, profil.php*)

- logowanie.php – logowanie klienta lub pracownika (z uwzględnieniem ról).
- rejestrowanie.php – formularz rejestracji nowego klienta z walidacją danych.
- profil.php – panel klienta z możliwością edycji danych i przeglądania konta.
- pracownicy_p.php – panel pracownika z dostępem do zamówień, reklamacji, kontaktów.
- Uwierzytelnianie oparte na sesji, z oddzielną obsługą klientów i personelu.

4 . Elementy Statyczne

4.1 HTML – struktura dokumentów

Strony projektu są napisane w HTML5 z wbudowanym kodem PHP. Główne sekcje każdej strony to:

- <header> – nagłówek z logo i menu,
- <nav> – pasek nawigacyjny z odnośnikami do podstron,
- <main> – główna zawartość strony (formularze, listy, dane z bazy),
- <footer> – stopka z informacjami kontaktowymi i prawami autorskimi.

Elementy takie jak formularze logowania, rejestracji, dodawania opinii i zamówień są zawarte w znacznikach <form>, z atrybutami required i method="POST".

4.2 CSS – stylizacja (*style.css*)

Wszystkie podstrony korzystają z jednego arkusza stylów *style.css*, w którym zdefiniowano klasy i identyfikatory do:

- pozycjonowania formularzy (*.form-box*, *.form-group*),
- stylowania tabel (*.table*, *.table-header*, *.table-row*),
- stylów przycisków (*.btn*, *.btn-success*, *.btn-danger*),
- wyglądu panelu administracyjnego (*.admin-panel*, *.content-box*).

Kolorystyka oparta jest o odcienie czerwieni (#8B0000), bieli i szarości.

Stylizacja jest prosta i intuicyjna, bez frameworków typu Bootstrap. Strona jest częściowo responsywna.

Strona_glowna.php

```
<!DOCTYPE html>
<html lang="pl">

<head>
    <!-- Podstawowe meta tagi -->
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>MeatMaster - Hurtownia Mięsa</title>

    <!-- Podpięcie arkuszy stylów -->
    <link rel="stylesheet" href="style.css"> <!-- Główny arkusz stylów -->
    <link rel="icon" type="image/png" href="icon.png"> <!-- Favicon strony -->

    <!-- Biblioteka ikon Font Awesome -->
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">
</head>

<body>
    <!-- Nagłówek strony z logo i menu nawigacyjnym -->
    <header>
        <div class="kontener naglowek-kontener">
            <!-- Logo firmy -->
            <div class="logo">
                
            </div>

            <!-- Główne menu nawigacyjne -->
            <nav>
                <ul>
                    <!-- Standardowe linki nawigacyjne -->
                    <li><a href="Strona_glowna.php">Strona główna</a></li>
                    <li><a href="Oferta.php">Oferta</a></li>
                    <li><a href="sklep.php">Sklep</a></li>
                    <li><a href="o_nas.php">O nas</a></li>
                    <li><a href="kontakt.php">Kontakt</a></li>
                    <li><a href="faq.php">FAQ</a></li>
                    <li><a href="aktualnosci.php">Aktualności</a></li>
                    <li><a href="opinie.php">Opinie</a></li>
                </ul>
            </nav>
        </div>
    </header>
</body>
</html>
```

```

        </ul>
      </div>
    </div>
  </header>

  <!-- Sekcja hero z dużym zdjęciem w tle i przyciskami CTA -->
  <section class="sekcja-glowna">
    <div class="kontener zawartosc-sekcji">
      <h2>Świeże mięso do Twojego sklepu!</h2>
      <p>Najwyższej jakości produkty mięsne od sprawdzonych dostawców.</p>

      <!-- Kontener z przyciskami Call-to-Action -->
      <div class="kontener-przyciskow">
        <a href="Oferta.php" class="przycisk">Oferta</a>
        <a href="sklep.php" class="przycisk">Sklep</a>
        <a href="faq.php" class="przycisk">FAQ</a>
        <a href="kontakt.php" class="przycisk">Kontakt</a>
      </div>
    </div>
  </section>

  <!-- Sekcja "O nas" z tekstem i zdjęciem -->
  <section class="sekcja-o-nas">
    <div class="kontener">
      <h2 class="tytul-sekcji">Hurtownia MeatMaster</h2>

      <div class="zawartosc-o-nas">
        <!-- Tekst informacyjny o firmie -->
        <div class="tekst-o-nas">
          <p>Jesteśmy liderem w dostawach wysokiej jakości mięsa dla sklepów i restauracji w całym kraju. Nasza hurtownia działa na rynku od ponad 15 lat, zdobywając zaufanie setek zadowolonych klientów.</p>
          <p>Współpracujemy wyłącznie ze sprawdzonymi hodowcami i dostawcami, co gwarantuje najwyższą jakość naszych produktów. Wszystkie mięsa przechodzą rygorystyczne kontrole weterynaryjne.</p>
          <p>Oferujemy konkurencyjne ceny, regularne dostawy i profesjonalne doradztwo w doborze asortymentu dla Twojego biznesu.</p>
        </div>

        <!-- Zdjęcie ilustracyjne -->
        <div class="obraz-o-nas">
          
        </div>
      </div>
    </div>
  </section>

  <!-- Sekcja z opiniami klientów -->
  <section class="sekcja-opinie">
    <div class="kontener">
      <h2 class="tytul-sekcji">Opinie naszych klientów</h2>

      <!-- Siatka z kartami opinii -->
      <div class="siatka-opinii">
        <!-- Opinia 1 -->
        <div class="karta-opinii">
          <div class="tresc-opinii">
            <p>Współpracujemy z MeatMaster od wczoraj i nigdy nie zawiedliśmy się na jakości ich produktów. Świeże mięso, terminowe dostawy i profesjonalna obsługa to ich znaki rozpoznawcze.</p>
          </div>
          <div class="autor-opinii">Bonifacy Boiler, właściciel kebabowni "Kula mocy u Bonifacego"</div>
        </div>

        <!-- Opinia 2 -->
        <div class="karta-opinii">
          <div class="tresc-opinii">
            <p>Jako restauracja premium stawiamy tylko na najlepsze składniki. MeatMaster zawsze dostarcza mięso najwyższej jakości, które zachwycą naszych gości. Polecam z czystym sumieniem!</p>
          </div>
          <div class="autor-opinii">Anna Nowak, szefowa kuchni "Bistro Oby piątekza 🍷"</div>
        </div>
      </div>
    </div>
  </section>

  <!-- Stopka strony -->
  <footer>
    <div class="kontener">
      <div class="zawartosc-stopki">
        <!-- Kolumna z danymi kontaktowymi -->
        <div class="kolumna-stopki">
          <h3>Kontakt</h3>
          <p><i class="fas fa-map-marker-alt"></i> ul. Mięsna 14, 69-420 Radomyśl Wielki</p>
          <p><i class="fas fa-phone"></i> +48 694 202 137</p>
          <p><i class="fas fa-envelope"></i> kontaktujSieWariacieEssa@meatmaster.pl</p>
        </div>

        <!-- Kolumna z godzinami otwarcia -->
        <div class="kolumna-stopki">
          <h3>Godziny otwarcia</h3>
          <p>Pon-Pt: 6:00 - 22:00</p>
          <p>Sob: 7:00 - 14:00</p>
          <p>Niedz: Zamknięte</p>
        </div>

        <!-- Kolumna z mediami społecznościowymi -->
        <div class="kolumna-stopki">
          <h3>Śledź nas</h3>
          <div class="linki-spoiecznosciowe">
            <!-- Twitter (X) -->
            <a href="#" aria-label="Twitter" class="x-icon">X</a>
            <!-- TikTok -->
            <a href="#" aria-label="TikTok"><i class="fab fa-tiktok"></i></a>
            <!-- Instagram -->
            <a href="#" aria-label="Instagram"><i class="fab fa-instagram"></i></a>
          </div>
        </div>
      </div>

      <!-- Informacja o prawach autorskich -->
      <div class="prawa-autorskie">
        <p>© 2025 MeatMaster - Hurtownia Mięsa. Wszelkie prawa zastrzeżone.</p>
      </div>
    </div>
  </footer>
</body>
</html>

```

Style.css

```

* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  /* Ustawia obliczanie szerokości i wysokości z uwzględnieniem paddingu i obramowania */
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
}

body {
  background: #f9f9f9;
  color: #333;
  line-height: 1.6;
  /* Zwiększa odstęp między liniami tekstu dla lepszej czytelności */
}

.kontener {
  max-width: 1200px;
  margin: 0 auto;
  /* Wyśrodkowuje element poziomo */
  padding: 0 20px;
}

/* === Header === */
header {
  background: #fff;
  box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
  /* Dodaje cień pod nagłówkiem */
  position: sticky;
  /* Przykleja nagłówek do górnej krawędzi ekranu podczas przewijania */
  top: 0;
  z-index: 100;
  /* Zapewnia, że nagłówek będzie na wierzchu względem innych elementów */
}

.naglowek-kontener {
  display: flex;
  justify-content: space-between;
  align-items: center;
  padding: 15px 0;
  flex-wrap: wrap;
  /* Umożliwia zawijanie elementów, np. na mniejszych ekranach */
}

.logo img {
  height: 50px;
  margin-right: 10px;
}

nav ul {
  display: flex;
  list-style: none;
  gap: 20px;
  /* Odstęp między elementami listy */
  justify-content: center;
}

nav a {
  text-decoration: none;
  color: #333;
  font-weight: 600;
  transition: 0.3s;
  /* Płynna zmiana stylu podczas najechania kursorem */
}

nav a:hover {
  color: #c00;
  /* Zmiana koloru linku przy najechaniu */
}

```

```

/* === Hero === */
.sekcja-glowna {
  background: linear-gradient(rgba(0, 0, 0, 0.5), rgba(0, 0, 0, 0.5)), url('miecho_tlo.jpg') center/cover;
  /* Nakłada półprzezroczysty gradient na tło obrazkowe */
  height: 500px;
  display: flex;
  align-items: center;
  text-align: center;
  color: #fff;
  padding: 0 20px;
}

.sekcja-glowna h2 {
  font-size: 48px;
  margin-bottom: 20px;
  text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5);
  /* Cień tekstu dla lepszej widoczności na tle */
}

.sekcja-glowna p {
  font-size: 20px;
  max-width: 700px;
  margin: 0 auto 30px;
}

.kontener-przyciskow {
  display: flex;
  justify-content: center;
  flex-wrap: wrap;
  /* Przycisk może zawijać się do nowej linii */
  gap: 15px;
  /* Odstęp między przyciskami */
}

.przycisk {
  padding: 12px 30px;
  background: #c00;
  color: #fff;
  border: 2px solid #c00;
  border-radius: 4px;
  text-decoration: none;
  font-weight: 600;
  transition: 0.3s;
  /* Płynna animacja podczas najechania */
  text-align: center;
}

.przycisk:hover {
  background: transparent;
  /* Przycisk staje się przezroczysty */
  border-color: #fff;
  color: #fff;
  /* Zmiana koloru tekstu na biały przy najechaniu */
}

/* === Sekcja "O nas" === */
.sekcja-o-nas,
.sekcja-opinie {
  padding: 80px 0;
}

.sekcja-o-nas {
  background: #fff;
}

.tytul-sekcji {
  text-align: center;
  color: #c00;
  font-size: 36px;
  margin-bottom: 50px;
}

```

```

.tytul-sekcji::after {
  content: "";
  /* Tworzy dodatkowy element graficzny (czerwony pasek pod nagłówkiem) */
  display: block;
  width: 80px;
  height: 3px;
  background: #c00;
  margin: 15px auto 0;
}

.zawartosc-o-nas {
  display: flex;
  gap: 40px;
  flex-wrap: wrap;
  /* Zawijanie elementów, np. na małych ekranach */
  align-items: center;
}

.tekst-o-nas {
  flex: 1;
}

.tekst-o-nas p {
  margin-bottom: 20px;
  font-size: 16px;
}

.obraz-o-nas {
  flex: 1;
  border-radius: 8px;
  overflow: hidden;
  /* Ukrywa wszystko, co wychodzi poza ramkę obrazka */
  box-shadow: 0 5px 15px rgba(0, 0, 0, 0.1);
  /* Dodaje cień wokół obrazka */
}

.obraz-o-nas img {
  width: 100%;
  display: block;
}

/* === Opinie === */
.sekcja-opinie {
  background: #f5f5f5;
}

.siatka-opinii {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));
  /* Tworzy responsywną siatkę kart opinii */
  gap: 30px;
}

.karta-opinii {
  background: #fff;
  padding: 30px;
  border-radius: 8px;
  box-shadow: 0 5px 15px rgba(0, 0, 0, 0.05);
  /* Dodaje delikatny cień do kart opinii */
}

.karta-opinii::before {
  font-size: 60px;
  color: rgba(204, 0, 0, 0.1);
  position: absolute;
  top: 20px;
  left: 20px;
  font-family: Georgia, serif;
  /* Przygotowanie do dodania ozdobnego elementu, np. cudzysłowu, ale brakuje "content:" */
}

```



```

.tresc-opinii {
  font-style: italic;
  margin-bottom: 20px;
  position: relative;
  /* Wymagane do użycia pseudo-elementów ::before lub ::after */
}

.autor-opinii {
  font-weight: 600;
  color: #c00;
}

/* === Stopka === */
footer {
  background: #222;
  color: #bbb;
  padding: 60px 0 20px;
}

.zawartosc-stopki {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
  /* Stopka z kolumnami, które same dostosowują szerokość */
  gap: 40px;
  margin-bottom: 40px;
}

.kolumna-stopki h3 {
  color: #c00;
  margin-bottom: 20px;
  font-size: 20px;
}

.kolumna-stopki p,
.kolumna-stopki a {
  margin-bottom: 10px;
  color: #bbb;
  text-decoration: none;
  transition: 0.3s;
  /* Płynna zmiana koloru przy najechaniu */
}

.kolumna-stopki a:hover {
  color: #fff;
  /* Zmiana koloru linku na biały przy najechaniu */
}

.linki-spoecznościowe {
  display: flex;
  gap: 15px;
}

.linki-spoecznościowe a {
  width: 40px;
  height: 40px;
  background: #c00;
  color: #fff;
  border-radius: 50%;
  /* Tworzy okrągłe przyciski */
  display: flex;
  align-items: center;
  justify-content: center;
  transition: 0.3s;
  /* Płynna animacja zmiany koloru */
}

.prawa-autorskie {
  text-align: center;
  border-top: 1px solid #444;
  /* Linia oddzielająca tekst praw autorskich */
  padding-top: 20px;
  font-size: 14px;
}

.contact-form {
  display: flex;
  flex-direction: column;
  align-items: center;
}

.form-group {
  display: flex;
  flex-direction: column;
  align-items: flex-start;
  margin-bottom: 15px;
  width: 100%;
  max-width: 400px;
}

.form-group label {
  margin-bottom: 5px;
}

input[type="text"],
textarea {
  width: 100%;
  padding: 8px;
  box-sizing: border-box;
}

button[type="submit"] {
  padding: 10px 20px;
  background-color: #c00;
  color: #fff;
  border: none;
  cursor: pointer;
  font-size: 16px;
  margin-top: 10px;
}

.kontakt-osoba {
  text-align: center;
  padding: 25px;
}

.kontakt-avatar {
  width: 120px;
  height: 120px;
  background-color: #c00;
  border-radius: 50%;
  margin: 0 auto 20px;
  display: flex;
  align-items: center;
  justify-content: center;
  color: #fff;
  font-size: 48px;
  font-weight: bold;
}

```

```

.kontakt-ikona {
  font-size: 48px;
}

.kontakt-dane {
  margin: 10px 0;
}

.kontakt-formularz {
  max-width: 800px;
  margin: 50px auto 0;
  background: #fff;
  padding: 30px;
  border-radius: 8px;
  box-shadow: 0 2px 10px rgba(0,0,0,0.1);
}

.kontakt-formularz h3 {
  text-align: center;
  color: #c00;
  margin-bottom: 30px;
}

.kontakt-formularz select {
  width: 100%;
  padding: 8px;
  margin-bottom: 15px;
}

```

5. JavaScript

5.1 Dynamiczne menu rozwijane

W projekcie zastosowano rozwijane menu nawigacyjne obsługiwane przez JavaScript (plik sklep.js). Menu reaguje na działania użytkownika, ułatwiając nawigację po stronie.

5.3 Obsługa przycisków i efektów interaktywnych

JavaScript wspiera dynamiczne akcje, takie jak dodawanie produktów do koszyka, aktualizacje ilości oraz zmiany stylu elementów po kliknięciu użytkownika.

```

// Prosty skrypt do filtrowania i sortowania produktów
document.addEventListener('DOMContentLoaded', function() {
    // 1. POBRANIE ELEMENTÓW STRONY
    const selectKategoria = document.getElementById('kategoria'); // Select z kategoriami
    const selectSortowanie = document.getElementById('sortowanie'); // Select z sortowaniem
    const kontener = document.querySelector('.siatka-produktow'); // Kontener na produkty
    const produkty = Array.from(document.querySelectorAll('.karta-produktu')); // Wszystkie produkty jako tablica

    // 2. GŁÓWNA FUNKCJA AKTUALIZUJĄCA WIDOK
    function aktualizujWidok() {
        // Pobieramy wybrane wartości
        const kategoria = selectKategoria.value;
        const sortowanie = selectSortowanie.value;

        // 2.1. FILTROWANIE PRODUKTÓW
        let przefiltrowane = produkty.filter(produkt => {
            if (kategoria === 'wszystkie') return true; // Pokazuj wszystkie produkty

            // Specjalna obsługa kategorii "kebab" (obejmuje też "kebab-drobiowe")
            if (kategoria === 'kebab') {
                return produkt.dataset.kategoria === 'kebab' ||
                    produkt.dataset.kategoria === 'kebab-drobiowe';
            }

            // Dla innych kategorii - dokładne dopasowanie
            return produkt.dataset.kategoria === kategoria;
        });

        // 2.2. SORTOWANIE PRODUKTÓW
        if (sortowanie === 'cena-roznaco') {
            przefiltrowane.sort(sortujCenaRoznaco);
        } else if (sortowanie === 'cena-malejaco') {
            przefiltrowane.sort(sortujCenaMalejaco);
        } else if (sortowanie === 'nazwa') {
            przefiltrowane.sort(sortujNazwa);
        }

        // 2.3. WYŚWIETLANIE WYNIKÓW
        kontener.innerHTML = ''; // Wyczyszczenie kontenera
        przefiltrowane.forEach(p => kontener.appendChild(p)); // Dodanie przefiltrowanych produktów
    }

    // 3. FUNKCJE SORTUJĄCE
    function sortujCenaRoznaco(a, b) {
        return pobierzCene(a) - pobierzCene(b); // Sortowanie od najtańszego
    }

    function sortujCenaMalejaco(a, b) {
        return pobierzCene(b) - pobierzCene(a); // Sortowanie od najdroższego
    }

    function sortujNazwa(a, b) {
        const nazwaA = a.querySelector('.nazwa-produktu').textContent.toLowerCase();
        const nazwaB = b.querySelector('.nazwa-produktu').textContent.toLowerCase();
        return nazwaA.localeCompare(nazwaB); // Sortowanie alfabetyczne
    }

    // 4. FUNKCJA POMOCNICZA DO POBRANIA CENY
    function pobierzCene(produkt) {
        const cenaText = produkt.querySelector('.cena-produktu').textContent;
        // Konwersja tekstu ceny na liczbę (zamiana , na . i usunięcie "zł/kg")
        const cena = parseFloat(cenaText.replace(' zł/kg', '').replace(',', '.'));
        return isNaN(cena) ? 0 : cena; // Dla "Zapytaj o ofertę" zwracamy 0
    }

    // 5. NASTUCHIWANIE ZDARZEŃ
    selectKategoria.addEventListener('change', aktualizujWidok); // Zmiana kategorii
    selectSortowanie.addEventListener('change', aktualizujWidok); // Zmiana sortowania

    // 6. INICJALIZACJA - PIERWSZE WYŚWIETLENIE
    aktualizujWidok();
});

```

6. Elementy dynamiczne (PHP)

6.1 Obsługa logowania i rejestracji użytkownika

Pliki logowanie.php i rejestrowanie.php obsługują logowanie klientów i pracowników z wykorzystaniem sesji i bezpiecznego haszowania haseł.

Logowanie.php

```
// 1. INICJALIZACJA SESJI - wymagana do przechowywania stanu logowania
session_start();

// 2. DOŁĄCZENIE PLIKU Z POŁĄCZENIEM DO BAZY DANYCH
require_once "db.php"; // Plik powinien zawierać zmienną $conn z połączeniem

// 3. INICJALIZACJA ZMIENNEJ NA KOMUNIKATY
$komunikat = ""; // Będzie przechowywał komunikaty o błędach

// 4. OBSŁUGA FORMULARZA LOGOWANIA (tylko dla żądań POST)
if ($_SERVER["REQUEST_METHOD"] === "POST") {

    // 5. POBRANIE I WALIDACJA DANYCH Z FORMULARZA
    $email = trim($_POST['email']); // Usunięcie białych znaków
    $haslo = trim($_POST['haslo']); // Usunięcie białych znaków

    // 6. SPRAWDZENIE CZY POLA NIE SĄ PUSTE
    if (empty($email) || empty($haslo)) {
        $komunikat = "Wprowadź email i hasło.";
    } else {

        // 7. PRÓBA LOGOWANIA

        // Zmienne na dane użytkownika
        $user = null;
        $rola = null;

        // 8. SPRAWDZENIE PRACOWNIKA
        $stmt = $conn->prepare("SELECT * FROM pracownicy WHERE email = ?");
        $stmt->bind_param("s", $email); // Zabezpieczenie przed SQL injection
        $stmt->execute();
        $result = $stmt->get_result();
        $pracownik = $result->fetch_assoc(); // Pobranie wyników

        // 9. WERYFIKACJA HASEŁ PRACOWNIKA
        if ($pracownik && $pracownik['haslo'] === $haslo) {
            $user = $pracownik;
            $rola = "pracownik";
        } else {

            // 10. SPRAWDZENIE KLIENTA (jeśli nie znaleziono pracownika)
            $stmt = $conn->prepare("SELECT * FROM klienci WHERE email = ?");
            $stmt->bind_param("s", $email);
            $stmt->execute();
            $result = $stmt->get_result();
            $klient = $result->fetch_assoc();

            // 11. WERYFIKACJA HASEŁ KLIENTA
            if ($klient && $klient['haslo'] === $haslo) {
                $user = $klient;
                $rola = "klient";
            }
        }

        // 12. JEŚLI ZNALEZIONO UŻYTKOWNIKA
        if ($user) {

            // 13. USTANOWIENIE DANYCH SESYJNYCH
            $_SESSION['zalogowany'] = true; // Flaga zalogowania
            $_SESSION['rola'] = $rola; // Rola (pracownik/klient)
            $_SESSION['user_id'] = $user['id'];
            $_SESSION['user_email'] = $user['email'];
            $_SESSION['imie'] = $user['imie'];
            $_SESSION['nazwisko'] = $user['nazwisko'];

            // 14. DODATKOWE DANE W ZALEŻNOŚCI OD ROLI
            if ($rola === 'pracownik') {
                $_SESSION['stanowisko'] = $user['stanowisko']; // Stanowisko pracownika
            } elseif ($rola === 'klient') {
                $_SESSION['typ_konta'] = $user['typ_konta']; // Typ konta klienta
            }

            // 15. PRZEKIEROWANIE PO ZALOGOWANIU
            header("Location: Strona_glowna.php");
            exit();
        } else {

            // 16. KOMUNIKAT O BŁĘDNYM LOGOWANIU
            $komunikat = "Nieprawidłowy email lub hasło.";
        }
    }
}

?>
```

Rejestrowanie.php

```
<?php
// Włączenie wyświetlania wszystkich błędów PHP (tylko do celów developerskich)
error_reporting(E_ALL);
ini_set('display_errors', 1);

// Sprawdzamy czy formularz został wysłany metodą POST
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    // Nawiązanie połączenia z bazą danych MySQL
    $conn = new mysqli("localhost", "root", "", "meatmasters");

    // Obsługa błędu połączenia
    if ($conn->connect_error) {
        die("Błąd połączenia: " . $conn->connect_error);
    }

    // Funkcja czyszcząca dane wejściowe (usuwa białe znaki i konwertuje znaki specjalne)
    function clean($dane) {
        return htmlspecialchars(trim($dane));
    }

    // Pobieranie i czyszczenie danych z formularza:
    $imie = clean($_POST['imie']);
    $nazwisko = clean($_POST['nazwisko']);
    $email = clean($_POST['email']);
    // Hasła nie czyszcimy, aby nie modyfikować specjalnych znaków w hasle
    $haslo = $_POST['haslo'];
    $potwierdz_haslo = $_POST['potwierdz-haslo'];
    $telefon = clean($_POST['telefon']);
    $typ_formularza = clean($_POST['typ-konta']);

    // Mapowanie wartości z formularza na wartości ENUM w bazie danych
    $mapa_typow = [
        'klient' => 'klient indywidualny',
        'firma' => 'firma/hurtownia',
        'restauracja' => 'restauracja'
    ];

    // Ustalenie typu konta z domyślną wartością 'klient indywidualny'
    $typ_konta = $mapa_typow[$typ_formularza] ?? 'klient indywidualny';

    // Pobieranie opcjonalnych danych firmy (jeśli istnieją)
    $nazwa_firmy = !empty($_POST['nazwa-firmy']) ? clean($_POST['nazwa-firmy']) : null;
    $nip = !empty($_POST['nip']) ? clean($_POST['nip']) : null;
    // Sprawdzenie czy checkbox regulaminu jest zaznaczony
    $regulamin = isset($_POST['regulamin']);

    // Walidacja danych:
    if ($haslo !== $potwierdz_haslo) {
        $error = "Hasła nie są takie same!";
    } elseif (strlen($haslo) < 8) {
        $error = "Hasło musi mieć co najmniej 8 znaków!";
    } elseif (!$regulamin) {
        $error = "Musisz zaakceptować regulamin!";
    } else {
        // Przypisanie hasła bez haszowania (UMAGA: to jest niebezpieczne w prawdziwej aplikacji)
        $czyste_haslo = $haslo;

        // Sprawdzenie unikalności emaila w bazie danych
        $stmt = $conn->prepare("SELECT id FROM klienci WHERE email = ?");
        $stmt->bind_param("s", $email);
        $stmt->execute();
        $stmt->store_result();

        if ($stmt->num_rows > 0) {
            $error = "Ten adres email już istnieje!";
        } else {
            // Przygotowanie zapytania INSERT
            $stmt = $conn->prepare("INSERT INTO klienci
                (imie, nazwisko, email, haslo, telefon, typ_konta, nazwa_firmy, nip)
                VALUES (?, ?, ?, ?, ?, ?, ?, ?)");
            $stmt->bind_param("ssssssss", $imie, $nazwisko, $email, $czyste_haslo, $telefon, $typ_konta, $nazwa_firmy, $nip);

            // Wykonanie zapytania i przekierowanie po sukcesie
            if ($stmt->execute()) {
                header("Location: logowanie.php?registered=1");
                exit();
            } else {
                $error = "Błąd przy rejestracji: " . $stmt->error;
            }
        }
        $stmt->close();
    }
}

// Zamknięcie połączenia z bazą danych
$conn->close();
}
```

6.2 Pobieranie danych z bazy (reklamacje_p.php)

Plik reklamacje_p.php odpowiada za pobieranie danych z widoku reklamacje_widok w bazie danych i wyświetlanie ich w formie tabeli.

Użytkownicy o odpowiednich uprawnieniach (np. pracownik, właściciel) mogą przeglądać szczegóły zgłoszeń oraz edytować ich status i decyzję. Wszystkie operacje realizowane są z wykorzystaniem zapytań SQL i bezpiecznego mechanizmu prepare() w PHP, co zabezpiecza system przed atakami typu SQL Injection

```
// 2. POBRANIE REKLAMACJI Z BAZY
$query = "SELECT * FROM reklamacje_widok"; // Widok z danymi reklamacji
$result = $conn->query($query);
$reklamacje = $result->fetch_all(MYSQLI_ASSOC); // Pobierz wszystkie rekordy jako tablicę asocjacyjną
```

6.3 Obsługa formularza kontaktowego

Formularz w kontakt.php umożliwi przesłanie wiadomości, z automatycznym uzupełnieniem danych zalogowanego użytkownika.

```
<?php

/******
 * SEKCJA 1: INICJALIZACJA
 *****/

// Włączamy sesję - potrzebna do sprawdzenia czy użytkownik jest zalogowany
session_start();

// łączymy się z bazą danych MySQL
// Parametry: serwer, login, hasło, nazwa bazy
$db = new mysqli('localhost', 'root', '', 'meatmasters');

/******
 * SEKCJA 2: POBRANIE DANYCH UŻYTKOWNIKA
 * (tylko dla zalogowanych)
 *****/

// Domyślne puste wartości dla formularza
$dane_user = [
    'imie' => '', // Imię i nazwisko
    'email' => '' // Adres email
];

// Sprawdzamy czy użytkownik jest zalogowany
if (isset($_SESSION['zalogowany'])) {
    // Pobieramy dane użytkownika z bazy
    $result = $db->query("SELECT imie, nazwisko, email FROM klienci WHERE id = ($_SESSION['user_id'])");

    // Jeśli znaleziono użytkownika, aktualizujemy dane
    if ($row = $result->fetch_assoc()) {
        $dane_user['imie'] = $row['imie'] . ' ' . $row['nazwisko'];
        $dane_user['email'] = $row['email'];
    }
}

/******
 * SEKCJA 3: OBSŁUGA FORMULARZA
 * (gdy użytkownik kliknie "Wyślij")
 *****/

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    // Zabezpieczamy dane przed SQL Injection
    $imie = $db->real_escape_string($_POST['imie'] ?? '');
    $email = $db->real_escape_string($_POST['email'] ?? '');
    $wiadomosc = $db->real_escape_string($_POST['wiadomosc'] ?? '');

    // Sprawdzamy czy wymagane pola są wypełnione
    if (empty($imie) && empty($email) && empty($wiadomosc)) {
        // Ustalamy ID użytkownika (NULL dla niezalogowanych)
        $user_id = isset($_SESSION['user_id']) ? $_SESSION['user_id'] : 'NULL';

        // Wstawiamy wiadomość do bazy danych
        $db->query("INSERT INTO kontakty (klient_id, imie, email, wiadomosc)
            VALUES ($user_id, '$imie', '$email', '$wiadomosc')");

        // Ustawiamy komunikat w sesji
        $_SESSION['komunikat'] = $db->error ? 'Błąd wysyłania!' : 'Wysłano!';
    } else {
        $_SESSION['komunikat'] = 'Wypełnij wszystkie pola!';
    }

    // Przekierowujemy z powrotem do formularza
    header('Location: kontakt.php');
    exit;
}

/******
 * SEKCJA 4: WYŚWIETLANIE KOMUNIKATÓW
 * (jeśli istnieją w sesji)
 *****/

if (isset($_SESSION['komunikat'])) {
    echo '<div class="komunikat"> ' . $_SESSION['komunikat'] . '</div>';
    unset($_SESSION['komunikat']); // Usuwanie komunikatu po wyświetleniu
}
?>
<!--
```

6.4 Panel administracyjny (dodawanie/edycja/usuwanie wpisów)

Panel pracownika (pracownicy_p.php, towar_p.php) umożliwia zarządzanie zawartością systemu, zależnie od poziomu uprawnień użytkownika.

```
<?php

/**
 * PLIK: towary_p.php
 * PRZEZNACZENIE: Panel zarządzania asortymentem mięsnym (tylko dla kierownika)
 * DOSTĘP: Tylko dla użytkowników z rolą 'Kierownik'
 * BEZPIECZEŃSTWO: Weryfikacja sesji, ochrona przed SQL injection
 */

// -----
// SEKCJA 1: INICJALIZACJA I KONTROLA DOSTĘPU
// -----

// Załączenie mechanizmu sesji i kontroli dostępu
require_once "sesje.php";
// Weryfikacja czy użytkownik ma uprawnienia - właściciel lub kierownik
if (!czyWlasciciel() && !czyKierownik()) {
    header("Location: brak_dostepu.php");
    exit();
}

require_once "db.php";

// -----
// SEKCJA 2: OPERACJE NA BAZIE DANYCH
// -----

/**
 * Pobranie pełnej listy towarów z widoku bazy danych
 * Widok 'towary_widok' łączy dane z różnych tabel dla wygody prezentacji
 */
$query = "SELECT * FROM towary_widok";
$result = $conn->query($query);
$towary = $result->fetch_all(MYSQLI_ASSOC); // Konwersja wyników na tablicę asocjacyjną

/**
 * Obsługa żądania aktualizacji danych towaru
 * Zabezpieczenia:
 * - Weryfikacja metody POST
 * - Sprawdzenie czy wysłano formularz edycji
 * - Prepared statements chroniące przed SQL injection
 */
if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['edytuj'])) {
    // Pobranie i sanityzacja danych z formularza
    $id = $_POST['id']; // ID towaru do aktualizacji
    $nazwa = $_POST['nazwa']; // Nowa nazwa produktu
    $cena = $_POST['cena']; // Zaktualizowana cena
    $kategoria = $_POST['kategoria']; // Kategoria produktu
    $dostepnosc = $_POST['dostepnosc']; // Status dostępności

    // Przygotowanie zapytania SQL z parametrami
    $update_query = "UPDATE towary SET
        nazwa = ?,          -- Parametr 1: Nazwa towaru
        cena_zl_kg = ?,     -- Parametr 2: Cena za kg
        kategoria = ?,      -- Parametr 3: Kategoria
        dostepnosc = ?      -- Parametr 4: Status dostępności
    WHERE id = ?";

    // Inicjalizacja prepared statement
    $stmt = $conn->prepare($update_query);

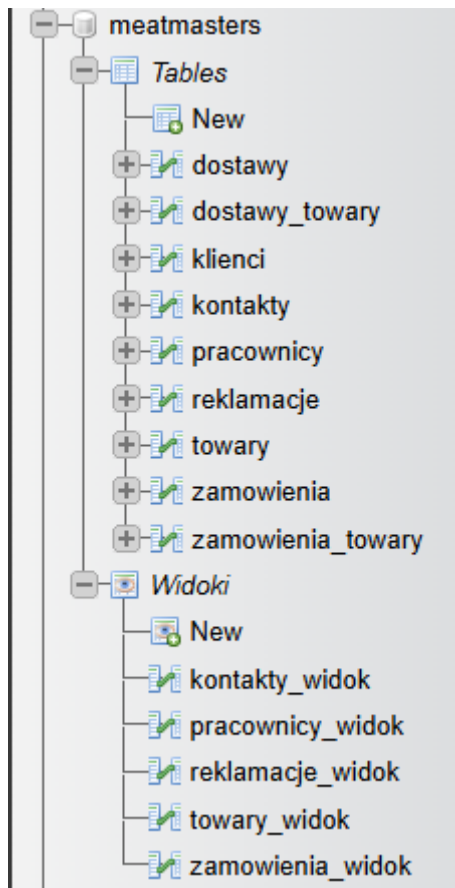
    // Powiązanie parametrów (typów):
    // s - string (nazwa)
    // d - double (cena)
    // s - string (kategoria)
    // s - string (dostepnosc)
    // i - integer (id)
    $stmt->bind_param("sdssi", $nazwa, $cena, $kategoria, $dostepnosc, $id);

    // Wykonanie zapytania
    $stmt->execute();

    // Obsługa rezultatów operacji
    if ($stmt->affected_rows > 0) {
        // Sukces - odświeżenie danych
        $success = "Dane towaru zostały zaktualizowane!";
        $result = $conn->query($query);
        $towary = $result->fetch_all(MYSQLI_ASSOC);
    } else {
        // Błąd aktualizacji
        $error = "Operacja aktualizacji nie powiodła się!";
    }
}
?>
```


7. Bazy danych

I. Projekt struktury bazy danych:



II. Opis tabel i pól

Tabele główne:

- **klienci** – przechowuje informacje o klientach (indywidualnych, firmach, restauracjach), w tym dane kontaktowe, typ konta, NIP i datę rejestracji.
- **pracownicy** – zawiera dane pracowników firmy, takie jak imię, nazwisko, e-mail, stanowisko, telefon, wynagrodzenie oraz data zatrudnienia.

- **towary** – baza produktów mięsnych, z informacjami o nazwie, cenie, kategorii, dostępności i przypisanym pracownikowi.
 - **zamowienia** – przechowuje szczegóły dotyczące zamówień składanych przez klientów, w tym dane adresowe, asortyment, wagę i status realizacji.
 - **zamowienia_towary** – tabela pośrednia zawierająca szczegóły, jakie towary i w jakich ilościach znalazły się w konkretnym zamówieniu.
 - **dostawy** – informacje o dostawach surowców od zewnętrznych dostawców: data, faktura, status i pracownik przyjmujący.
 - **dostawy_towary** – szczegółowe dane o towarach dostarczonych w ramach poszczególnych dostaw: ilość i cena zakupu.
 - **reklamacje** – obsługuje proces reklamacji składanych przez klientów, wraz z opisem problemu, statusem, decyzją i osobą rozpatrującą.
 - **kontakty** – wiadomości i zgłoszenia od klientów, przypisane do odpowiedniego tematu (zamówienie, reklamacja, współpraca, inne) i pracownika.
-

Widoki (widzialne zestawienia danych):

- **kontakty_widok** – uproszczony podgląd kontaktów od klientów z danymi nadawcy, tematami i przypisanym pracownikiem.
- **zamowienia_widok** – widok prezentujący szczegółowe informacje o zamówieniach w formacie gotowym do raportowania (łącznie z klientem, firmą, adresem itp.).
- **towary_widok** – prezentuje listę towarów z kategorią, dostępnością, ceną oraz nazwiskiem opiekuna produktu.
- **pracownicy_widok** – uproszczona lista pracowników z podstawowymi danymi i wynagrodzeniem.
- **reklamacje_widok** – prezentuje reklamacje wraz z informacjami o kliencie, zamówieniu oraz pracowniku rozpatrującym.

III. relacje

zamowienia.klient_id → klienci.id – tabela zamowienia ma klucz obcy klient_id, który wskazuje, kto złożył dane zamówienie.

zamowienia_towary.zamowienie_id → zamowienia.id,
zamowienia_towary.towar_id → towary.id – tabela
zamowienia_towary łączy konkretne towary z konkretnymi
zamówieniami (relacja wiele-do-wielu).

dostawy.pracownik_id → pracownicy.id – tabela dostawy ma klucz obcy pracownik_id, który określa pracownika odpowiedzialnego za dostawę.

dostawy_towary.dostawa_id → dostawy.id,
dostawy_towary.towar_id → towary.id – tabela dostawy_towary
łączy konkretne towary z dostawami, w których zostały dostarczone.

towary.pracownik_odpowiedzialny → pracownicy.id – tabela towary
zawiera klucz obcy wskazujący, który pracownik odpowiada za dany
produkt.

reklamacje.zamowienie_id → zamowienia.id, reklamacje.klient_id →
klienci.id, reklamacje.pracownik_id → pracownicy.id – tabela
reklamacje zawiera informacje o reklamacji konkretnego zamówienia,
złożonej przez danego klienta i rozpatrywanej przez pracownika.

kontakty.klient_id → klienci.id, kontakty.pracownik_id →
pracownicy.id – tabela kontakty łączy klienta, który wysłał zgłoszenie,
z pracownikiem, który je obsługuje.

IV. Przykładowe zapytania SQL

```
SELECT * FROM produkty WHERE cena < 50;
```

```
INSERT INTO kontakty (klient_id, imie, email, wiadomosc)  
VALUES (1, 'Jan', 'jan@wp.pl', 'Zapytanie o ofertę');
```

8. Bezpieczeństwo

8.1. Walidacja danych wejściowych

W celu ochrony aplikacji przed błędami i atakami (np. SQL Injection, XSS), wszystkie dane wprowadzane przez użytkownika powinny być walidowane:

- Na poziomie **frontend** – np. przy pomocy HTML5, JavaScript (sprawdzanie długości, typu danych),
- Na poziomie **backend** – filtrowanie danych w PHP, użycie przygotowanych zapytań (prepared statements),
- Dla pól typu e-mail, telefon czy NIP stosuje się dodatkowe wzorce walidacyjne (regex).

8.2. Obsługa sesji i uwierzytelnianie

- System logowania oparty jest na **mechanizmie sesji** (session_start() w PHP).
- Hasła użytkowników są przechowywane w formie zaszyfrowanej (np. password_hash()).
- Dostęp do panelu administratora/personelu wymaga autoryzacji – użytkownicy z odpowiednimi uprawnieniami są kierowani na dedykowane widoki.
- Po stronie serwera stosowana jest kontrola dostępu do poszczególnych zasobów (sprawdzanie ról i session['user_id']).

9. Możliwości rozbudowy

9.1. Dodanie wyszukiwarki

Możliwość rozszerzenia systemu o funkcję wyszukiwania np. produktów, klientów, zamówień. Wyszukiwarka mogłaby wykorzystywać zapytania typu LIKE, filtrowanie według kategorii, statusów itp.

9.2. System ocen i komentarzy

Wprowadzenie modułu umożliwiającego klientom ocenianie i komentowanie towarów lub usług. Dane te mogłyby wpływać na średnią ocenę produktu.

9.3. System rezerwacji online

Możliwość zarezerwowania towaru z wyprzedzeniem, np. na konkretny dzień lub godzinę. Taki system wymagałby tabeli rezerwacje oraz integracji z aktualną dostępnością towarów.

9.4. Responsywność i aplikacja mobilna

Dalszy rozwój mógłby objąć:

- Pełne dostosowanie strony do urządzeń mobilnych (RWD),
- Stworzenie aplikacji mobilnej dla systemów Android/iOS – np. do składania zamówień, śledzenia reklamacji czy kontaktu z obsługą.

10.1

```
<?php
// Włączenie wyświetlania wszystkich błędów PHP (tylko do celów developerskich)
error_reporting(E_ALL);
ini_set('display_errors', 1);

// Sprawdzamy czy formularz został wysłany metodą POST
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    // Nawiązanie połączenia z bazą danych MySQL
    $conn = new mysqli("localhost", "root", "", "meatmasters");

    // Obsługa błędu połączenia
    if ($conn->connect_error) {
        die("Błąd połączenia: " . $conn->connect_error);
    }

    // Funkcja czyszcząca dane wejściowe (usuwa białe znaki i konwertuje znaki specjalne)
    function clean($dane) {
        return htmlspecialchars(trim($dane));
    }

    // Pobieranie i czyszczenie danych z formularza:
    $imie = clean($_POST['imie']);
    $nazwisko = clean($_POST['nazwisko']);
    $email = clean($_POST['email']);
    // Hasła nie czyścimy, aby nie modyfikować specjalnych znaków w hasle
    $haslo = $_POST['haslo'];
    $potwierdz_haslo = $_POST['potwierdz-haslo'];
    $telefon = clean($_POST['telefon']);
    $typ_formularza = clean($_POST['typ-konta']);

    // Mapowanie wartości z formularza na wartości ENUM w bazie danych
    $mapa_typow = [
        'klient' => 'klient indywidualny',
        'firma' => 'firma/hurtownia',
        'restauracja' => 'restauracja'
    ];

    // Ustalenie typu konta z domyślną wartością 'klient indywidualny'
    $typ_konta = $mapa_typow[$typ_formularza] ?? 'klient indywidualny';

    // Pobieranie opcjonalnych danych firmy (jeśli istnieją)
    $nazwa_firmy = !empty($_POST['nazwa-firmy']) ? clean($_POST['nazwa-firmy']) : null;
    $nip = !empty($_POST['nip']) ? clean($_POST['nip']) : null;
    // Sprawdzenie czy checkbox regulaminu jest zaznaczony
    $regulamin = isset($_POST['regulamin']);

    // Walidacja danych:
    if ($haslo !== $potwierdz_haslo) {
        $error = "Hasła nie są takie same!";
    } elseif (strlen($haslo) < 8) {
        $error = "Hasło musi mieć co najmniej 8 znaków!";
    } elseif (!$regulamin) {
        $error = "Musisz zaakceptować regulamin!";
    } else {
        // Przypisanie hasła bez haszowania (UWAGA: to jest niebezpieczne w prawdziwej aplikacji)
        $czyste_haslo = $haslo;

        // Sprawdzenie unikalności emaila w bazie danych
        $stmt = $conn->prepare("SELECT id FROM klienci WHERE email = ?");
        $stmt->bind_param("s", $email);
        $stmt->execute();
        $stmt->store_result();

        if ($stmt->num_rows > 0) {
            $error = "Ten adres email już istnieje!";
        } else {
            // Przygotowanie zapytania INSERT
            $stmt = $conn->prepare("INSERT INTO klienci
                (imie, nazwisko, email, haslo, telefon, typ_konta, nazwa_firmy, nip)
                VALUES (?, ?, ?, ?, ?, ?, ?, ?)");
            $stmt->bind_param("ssssssss", $imie, $nazwisko, $email, $czyste_haslo, $telefon, $typ_konta, $nazwa_firmy, $nip);

            // Wykonanie zapytania i przekierowanie po sukcesie
            if ($stmt->execute()) {
                header("Location: logowanie.php?registered=1");
                exit();
            } else {
                $error = "Błąd przy rejestracji: " . $stmt->error;
            }
        }
        $stmt->close();
    }
}

// Zamknięcie połączenia z bazą danych
$conn->close();
}
```

Kod PHP odpowiada za proces rejestracji nowych klientów do systemu hurtowni. Główne kroki działania:

- Połączenie z bazą danych meatmasters (lokalna instancja MySQL).
- Odczytanie danych z formularza: imię, nazwisko, e-mail, telefon, hasło i jego potwierdzenie.
- Dla bezpieczeństwa: dane tekstowe są czyszczone funkcją `htmlspecialchars()` i `trim()`.
- Hasło nie jest modyfikowane przed sprawdzeniem, aby nie naruszyć składni specjalnej.
- Hasła są porównywane – jeśli się zgadzają, właściwe hasło jest szyfrowane funkcją `password_hash()` (algorytm domyślny, np. `bcrypt`).
- W przypadku zgodnych danych wykonywane jest bezpieczne zapytanie `INSERT` przy pomocy przygotowanego statementu (`prepare` + `bind_param`).
- Po udanym zapisie wyświetlany jest komunikat o sukcesie.

```

// 1. INICJALIZACJA SESJI - wymagana do przechowywania stanu logowania
session_start();

// 2. DOŁĄCZENIE PLIKU Z POŁĄCZENIEM DO BAZY DANYCH
require_once "db.php"; // Plik powinien zawierać zmienną $conn z połączeniem

// 3. INICJALIZACJA ZMIENNEJ NA KOMUNIKATY
$komunikat = ""; // Będzie przechowywał komunikaty o błędach

// 4. OBSŁUGA FORMULARZA LOGOWANIA (tylko dla żądań POST)
if ($_SERVER["REQUEST_METHOD"] === "POST") {

    // 5. POBRANIE I WALIDACJA DANYCH Z FORMULARZA
    $email = trim($_POST['email']); // Usunięcie białych znaków
    $haslo = trim($_POST['haslo']); // Usunięcie białych znaków

    // 6. SPRAWDZENIE CZY POLA NIE SĄ PUSTE
    if (empty($email) || empty($haslo)) {
        $komunikat = "Wprowadź email i hasło.";
    } else {

        // 7. PRÓBA LOGOWANIA

        // Zmienno na dane użytkownika
        $user = null;
        $rola = null;

        // 8. SPRAWDZENIE PRACOWNIKA
        $stmt = $conn->prepare("SELECT * FROM pracownicy WHERE email = ?");
        $stmt->bind_param("s", $email); // Zabezpieczenie przed SQL injection
        $stmt->execute();
        $result = $stmt->get_result();
        $pracownik = $result->fetch_assoc(); // Pobranie wyników

        // 9. WERYFIKACJA HASŁA PRACOWNIKA
        if ($pracownik && $pracownik['haslo'] === $haslo) {
            $user = $pracownik;
            $rola = "pracownik";
        } else {

            // 10. SPRAWDZENIE KLIENTA (jeśli nie znaleziono pracownika)
            $stmt = $conn->prepare("SELECT * FROM klienci WHERE email = ?");
            $stmt->bind_param("s", $email);
            $stmt->execute();
            $result = $stmt->get_result();
            $klient = $result->fetch_assoc();

            // 11. WERYFIKACJA HASŁA KLIENTA
            if ($klient && $klient['haslo'] === $haslo) {
                $user = $klient;
                $rola = "klient";
            }
        }

        // 12. JEŚLI ZNALEZIONO UŻYTKOWNIKA
        if ($user) {

            // 13. USTANOWIENIE DANYCH SESYJNYCH
            $_SESSION['zalogowany'] = true; // Flaga zalogowania
            $_SESSION['rola'] = $rola; // Rola (pracownik/klient)
            $_SESSION['user_id'] = $user['id'];
            $_SESSION['user_email'] = $user['email'];
            $_SESSION['imie'] = $user['imie'];
            $_SESSION['nazwisko'] = $user['nazwisko'];

            // 14. DODATKOWE DANE W ZALEŻNOŚCI OD ROLI
            if ($rola === 'pracownik') {
                $_SESSION['stanowisko'] = $user['stanowisko']; // Stanowisko pracownika
            } elseif ($rola === 'klient') {
                $_SESSION['typ_konta'] = $user['typ_konta']; // Typ konta klienta
            }

            // 15. PRZEKIEROWANIE PO ZALOGOWANIU
            header("Location: Strona_glowna.php");
            exit();
        } else {

            // 16. KOMUNIKAT O BŁĘDNYM LOGOWANIU
            $komunikat = "Nieprawidłowy email lub hasło.";
        }
    }
}
}
}

```

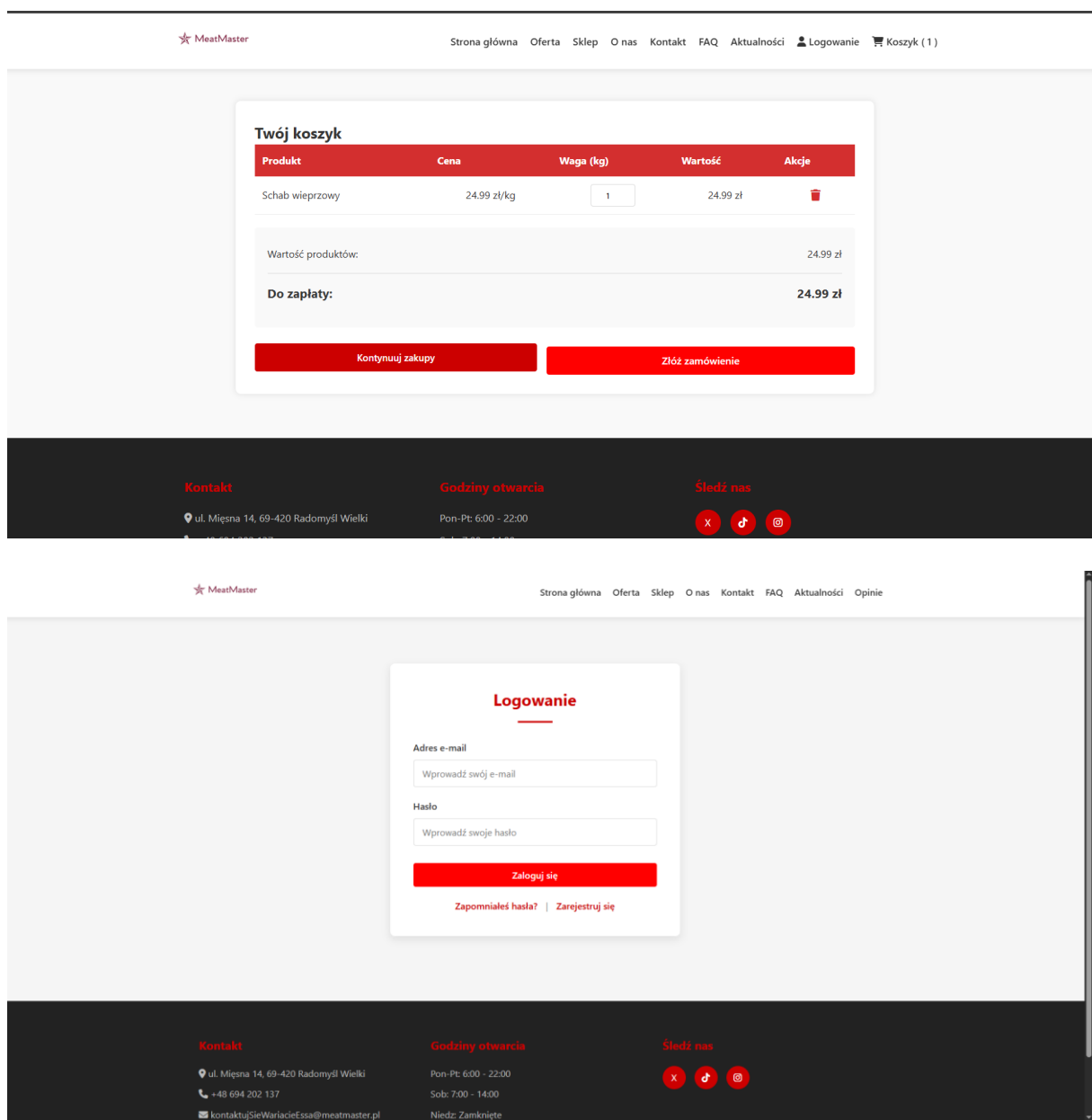

Plik obsługuje logowanie pracownika do systemu:

- Otwiera sesję (`session_start()`), żeby zapamiętać zalogowanego użytkownika.
- Łączy się z bazą danych (`db.php`).
- Pobiera e-mail i hasło z formularza logowania.
- Szuka pracownika w bazie po e-mailu (tabela `pracownicy`).
- Sprawdza, czy hasło się zgadza (funkcja `password_verify()`).
- Jeśli dane są poprawne:
 - Ustawia dane w sesji (`id`, `rola`, `zalogowany`).
 - Przekierowuje użytkownika do panelu (`pracownicy_p.php`).
- Jeśli dane są błędne – wyświetla odpowiedni komunikat (np. „Błędne hasło”).

Skrypt zabezpiecza logowanie przez:

- Szyfrowanie haseł,
- Przygotowane zapytania (`prepare()`), które chronią przed SQL Injectio

10.2 Zrzuty ekranu działania systemu



Rejestracja konta

Imię *

Wprowadź swoje imię

Nazwisko *

Wprowadź swoje nazwisko

Adres e-mail *

Wprowadź swój e-mail

Hasło *

Wprowadź hasło

Potwierdź hasło *

Powtórz hasło

Telefon kontaktowy *

Wprowadź numer telefonu

Typ konta *

-- Wybierz typ konta --

Dane firmy (wypełnij jeśli rejestrujesz konto firmowe)

Nazwa firmy

Wprowadź nazwę firmy

NIP

Wprowadź NIP

☐ Akceptuję [Regulamin](#) serwisu *

☐ Chcę otrzymywać newsletter z ofertami specjalnymi

Zarejestruj się

Masz już konto? [Zaloguj się](#)

Zarządzanie pracownikami

Dodaj nowego pracownika

ID	Imię	Nazwisko	Email	Telefon	Stanowisko	Zatrudniony	Wynagrodzenie	Akcje
1	Jan	Kowalski	j.kowalski@meatmasters.pl	501234567	Właściciel	15.03.2020	8,500.00 zł	Edytuj Usuń
2	Anna	Nowak	a.nowak@meatmasters.pl	502345678	Kierownik	12.05.2019	8,700.00 zł	Edytuj Usuń
3	Piotr	Wiśniewski	p.wisniewski@meatmasters.pl	503456789	Programista	20.06.2021	9,500.00 zł	Edytuj Usuń
4	Katarzyna	Dąbrowska	k.dabrowska@meatmasters.pl	504567890	Programista	10.01.2022	9,200.00 zł	Edytuj Usuń
5	Marek	Lewandowski	m.lewandowski@meatmasters.pl	505678901	Pracownik linii pakowania	15.07.2021	4,200.00 zł	Edytuj Usuń
6	Agnieszka	Wójcik	a.wojcik@meatmasters.pl	506789012	Pracownik linii pakowania	01.04.2022	4,100.00 zł	Edytuj Usuń
7	Tomasz	Kamiński	t.kaminski@meatmasters.pl	507890123	Pracownik linii pakowania	18.11.2020	4,300.00 zł	Edytuj Usuń
8	Magdalena	Zajac	m.zajac@meatmasters.pl	508901234	Pracownik linii pakowania	22.05.2021	4,250.00 zł	Edytuj Usuń
9	Grzegorz	Szymański	g.szymanski@meatmasters.pl	509012345	Pracownik linii pakowania	01.03.2022	4,150.00 zł	Edytuj Usuń
10	Joanna	Woźniak	j.wozniak@meatmasters.pl	511234567	Magazynier	10.08.2020	4,500.00 zł	Edytuj Usuń
11	Robert	Kozłowski	r.kozlowski@meatmasters.pl	512345678	Magazynier	30.09.2021	4,600.00 zł	Edytuj Usuń
12	Ewa	Jankowska	e.jankowska@meatmasters.pl	513456789	Magazynier	01.07.2022	4,550.00 zł	Edytuj Usuń
13	Paweł	Mazur	p.mazur@meatmasters.pl	514567890	Księgowy	10.02.2020	6,800.00 zł	Edytuj Usuń
14	Monika	Krawczyk	m.krawczyk@meatmasters.pl	515678901	Specjalista HR	15.04.2021	6,200.00 zł	Edytuj Usuń

Nr	Nazwa produktu	Cena (zł/kg)	Kategoria	Dostępność	Abyć
1	Rolbit woskowy	89,99	woskoina	dostępny	Kup
2	Wapno wierzynka	27,99	wapnowina	dostępny	Kup
3	Mieszanka do kebabu	0,00	mieszanka	zapasy	Kup
4	Mieszanka woskowa	34,99	woskoina	dostępny	Kup
5	Flak z kurczaka	25,99	drożdż	dostępny	Kup
6	Mięso do kebabu drożdżowe	0,00	drożdż	zapasy	Kup
7	Schab wierzynkowy	28,99	wapnowina	dostępny	Kup
8	Ułóż z kurczaka	18,99	drożdż	dostępny	Kup

10.4. Lista błędów/uwag i pomysłów na ulepszenia

- Brak resetowania hasła – warto dodać funkcję przypominania/zmiany hasła przez e-mail.,
- Brak paginacji w sklepie przy dużej liczbie produktów.,
- Panel administracyjny mógłby mieć bardziej rozbudowane statystyki i filtrowanie.,
- Formularze nie zawsze mają pełną walidację po stronie klienta (JS).,
- Brak testów automatycznych i walidacji dostępności (WCAG).,
- Możliwość rozbudowy systemu opinii o oceny gwiazdkowe.,
- Dodanie historii zamówień i powiadomień e-mail dla klientów.,
- Ulepszenie responsywności na starszych urządzeniach mobilnych.,
- Integracja z płatnościami online.,
- Dodanie panelu do zarządzania asortymentem i promocjami.