

Dokumentacja techniczna projektu "SklepOnline"

Wykonał:

Daniel Ostojski

Spis treści

1.Wstęp	3
2.Cel projektu	4
3.Technologie i narzędzia	5
4.Opis ogólny systemu	6
5.Struktura plików	7
6.Baza danych	8
7.Obsługa użytkownika	10
8.Ogłoszenia i zakup	11
9.Komunikator (czat)	12
10.Bezpieczeństwo i walidacja	13
11.Interfejs użytkownika	14
12.Testowanie	15
13.Możliwości rozwoju	16
14.Podsumowanie	17

1. Wstęp

Projekt "SklepOnline" został zrealizowany jako rezultat pracy zaliczeniowej z zakresu technologii webowych.

Głównym celem było opracowanie kompletnej, w pełni funkcjonalnej aplikacji ogłoszeniowo-sprzedażowej przy użyciu czystych technologii – bez wspomagania się frameworkami czy zewnętrznymi bibliotekami. Projekt zrealizowano lokalnie w środowisku XAMPP.

Inspiracją do stworzenia tego typu systemu była chęć odwzorowania funkcjonalności znanych serwisów ogłoszeniowych i platform e-commerce, takich jak OLX czy Allegro, lecz w uproszczonej, edukacyjnej wersji, z pełną kontrolą nad kodem źródłowym. Dzięki temu projekt nie tylko spełnia wymogi programowe, ale także stanowi solidne zaplecze do dalszego rozwoju i nauki.

Aplikacja umożliwia dodawanie ogłoszeń, przypisywanie ich do kategorii, przeglądanie dostępnych ofert, komunikację z użytkownikami, a także symulację płatności, w tym zakup ogłoszeń, po którym stają się one niedostępne dla innych użytkowników.

2. Cel projektu

Głównym celem SklepOnline jest umożliwienie efektywnej wymiany informacji o produktach i usługach między użytkownikami poprzez funkcjonalny interfejs WWW. Projekt spełnia podstawowe założenia aplikacji ogłoszeniowej z elementami e-commerce. Konkretnie, aplikacja umożliwia:

- rejestrację nowych użytkowników oraz logowanie,
- dodawanie i edycję ogłoszeń przez zarejestrowanych użytkowników,
- przeglądanie listy aktywnych ogłoszeń przez wszystkich odwiedzających,
- filtrowanie ogłoszeń po dostępnych kategoriach,
- przeglądanie szczegółów ogłoszenia z możliwością kontaktu ze sprzedawcą,
- symulację płatności z wyborem metody (BLIK, karta, przelew),
- komunikację między użytkownikami poprzez wbudowany czat AJAX.

System zaprojektowano tak, by był prosty w obsłudze, szybki w działaniu oraz możliwy do rozszerzania o dodatkowe funkcje. Dzięki modularnej budowie, każdy komponent systemu może być w przyszłości zastąpiony lub zaktualizowany bez ingerencji w pozostałą część aplikacji.

3. Technologie i narzędzia

Projekt wykorzystuje wyłącznie podstawowe technologie webowe. Główne komponenty środowiska to:

- **HTML5:** odpowiada za strukturę dokumentu. Formularze, nagłówki, tabele i elementy interakcji z użytkownikiem zostały opracowane z jego pomocą.
- **CSS3:** służy do estetycznego prezentowania treści. Wykorzystano media queries, selektory atrybutów oraz zmienne w trybie ciemnym.
- **JavaScript:** używany głównie do obsługi dynamicznego czatu. W przyszłości może być podstawą rozbudowanego frontendu.
- **AJAX (Fetch API):** zastosowany do aktualizacji wiadomości w czasie rzeczywistym bez przeładowania strony.
- **PHP 8.2:** rdzeń logiki serwerowej. Obsługuje połączenia z bazą, sesje, walidacje i routing.
- **MySQL (MariaDB):** relacyjna baza danych do przechowywania ogłoszeń, wiadomości, danych użytkowników i płatności.
- **XAMPP:** lokalny serwer Apache + MySQL do testów i uruchamiania.
- **phpMyAdmin:** graficzny interfejs do zarządzania strukturą i rekordami bazy danych.

Cały projekt został zbudowany bez frameworków co czyni go przejrzystym.

4. Opis ogólny systemu

SklepOnline został zaprojektowany jako prosta, modułarna aplikacja webowa typu ogłoszeniowego, rozszerzona o możliwość dokonywania zakupów i wymiany wiadomości między użytkownikami. System działa w środowisku lokalnym, bez dostępu do internetu, i w pełni wykorzystuje możliwości PHP oraz MySQL do przechowywania danych i obsługi logiki aplikacji.

Struktura logiczna systemu została podzielona na trzy podstawowe warstwy:

- **Warstwa prezentacji (front-end)** – odpowiedzialna za interfejs użytkownika (HTML, CSS, JavaScript),
- **Warstwa logiki aplikacji (PHP)** – przetwarza żądania, zarządza sesjami, wykonuje zapytania SQL,
- **Warstwa danych (MySQL)** – przechowuje trwałe informacje o użytkownikach, ogłoszeniach, wiadomościach i płatnościach.

Komunikacja pomiędzy tymi warstwami odbywa się w sposób prosty i bezpieczny. Formularze użytkownika przetwarzane są przez skrypty PHP, które następnie komunikują się z bazą danych za pomocą PDO. Część dynamiczna (czat) zrealizowana została przy użyciu AJAX i odpowiedzi w formacie JSON.

5. Struktura plików

Projekt posiada przejrzystą strukturę katalogów i plików, co pozwala łatwo zrozumieć logikę działania systemu:

/projekt/		
—	index.php	Przekierowanie do lista.php
—	lista.php	Główna strona z listą ogłoszeń
—	dodaj.php	Formularz dodawania ogłoszenia
—	ogloszenie.php	Szczegóły pojedynczego ogłoszenia
—	kup.php	Przycisk zakupu i logika oznaczenia kupione
—	platnosc.php	Formularz wyboru metody płatności
—	login.php	Formularz logowania
—	register.php	Formularz rejestracji
—	logout.php	Skrypt wylogowujący
—	czaty.php	Lista prowadzonych rozmów
—	czat.php	Szczegóły rozmowy (komunikator)
—	ajax_czat.php	Endpoint AJAX do zapisu wiadomości
—	ajax_pobierz.php	Endpoint AJAX do odczytu wiadomości
—	db.php	Połączenie z bazą danych
—	style.css	Plik stylów
—	uploads/	Folder z przesłanymi zdjęciami
—	header.php	Wspólny nagłówek z nawigacją

Podział plików odzwierciedla podział odpowiedzialności logicznej. Pliki `ajax_*.php` realizują funkcjonalność API. `header.php` jest komponentem wspólnym wykorzystywanym w większości stron.

6. Baza danych

Projekt wykorzystuje relacyjną bazę danych MySQL do przechowywania i przetwarzania danych. Baza zawiera cztery podstawowe tabele:

6.1. Tabela **users**

Zawiera dane użytkowników:

- `id`: klucz główny,
- `email`: adres e-mail (unikalny),
- `username`: nazwa wyświetlana,
- `password`: hasło (hash).

6.2. Tabela **ogłoszenia**

Przechowuje dane o wszystkich ogłoszeniach:

- `id`: klucz główny,
- `user_id`: relacja do użytkownika,
- `tytuł`: tytuł ogłoszenia,
- `cena`: cena produktu,
- `zdjecie`: ścieżka do pliku graficznego,
- `kategoria`: kategoria (elektronika, motoryzacja itd.),
- `kupione`: flaga logiczna (czy ogłoszenie zostało kupione).

6.3. Tabela wiadomosci

Odpowiada za komunikację między użytkownikami:

- `id`: identyfikator wiadomości,
- `nadawca_id`, `odbiorca_id`: użytkownicy,
- `ogloszenie_id`: powiązanie z ogłoszeniem,
- `tresc`: treść wiadomości,
- `czas`: data i godzina wysłania.

6.4. Tabela payments

Zawiera dane dotyczące przeprowadzonych transakcji:

- `id`: identyfikator płatności,
- `user_id`: kupujący,
- `ad_id`: ogłoszenie,
- `method`: metoda płatności (BLIK, karta, przelew),
- `created_at`: znacznik czasu.

Każda z tabel zawiera klucze główne, a tam gdzie wymagane – także klucze obce do zachowania integralności danych.

7. Obsługa użytkownika

Interakcja z użytkownikiem odbywa się za pomocą formularzy oraz sesji. Po rejestracji użytkownik może się zalogować, a dane sesji przechowywane są w zmiennej `$_SESSION`. System uwzględnia rozróżnienie między użytkownikiem zalogowanym i gościem, przez co możliwe jest wyświetlanie spersonalizowanej nawigacji oraz przycisków dostępu.

Formularze rejestracyjne oraz logowania walidują dane po stronie serwera i wymagają poprawnego formatu adresu e-mail oraz haseł o minimalnej długości. Hasła są zapisywane w bazie w formie zahashowanej przy użyciu `password_hash()`.

Po zalogowaniu użytkownik może dodawać ogłoszenia, kontaktować się ze sprzedającym i dokonywać symulowanych zakupów. Interfejs został zaprojektowany tak, by każdy z tych procesów był intuicyjny i nie wymagał przechodzenia przez wiele podstron. Każda strona posiada przejrzysty nagłówek, linki powrotu oraz przyciski działania.

8. Ogłoszenia i zakup

Dodawanie ogłoszeń odbywa się za pośrednictwem formularza dostępnego po zalogowaniu. Użytkownik wprowadza tytuł, cenę, wybiera kategorię oraz przesyła zdjęcie. Dane trafiają do bazy i są natychmiast widoczne na stronie głównej.

Na liście ogłoszeń wyświetlane są aktywne pozycje. Każda karta ogłoszenia zawiera miniaturkę zdjęcia, tytuł, cenę oraz nazwę użytkownika. Użytkownik ma możliwość kliknięcia w szczegóły oferty.

Po wejściu w szczegóły ogłoszenia pojawia się opcja „Kup teraz”. Kliknięcie w ten przycisk prowadzi do formularza płatności. Po wybraniu jednej z trzech metod płatności (BLIK, karta, przelew) i potwierdzeniu transakcji, ogłoszenie oznaczane jest jako kupione (pole `kupione=1`) i przestaje być wyświetlane na stronie głównej.

9. Komunikator (czat)

Komunikacja pomiędzy użytkownikami odbywa się poprzez zintegrowany czat. Moduł czatu obsługuje AJAX, dzięki czemu wiadomości aktualizują się automatycznie bez przeładowania strony.

Funkcjonalność obejmuje:

- możliwość wysyłania wiadomości z poziomu strony ogłoszenia,
- przegląd historii konwersacji przypisanej do konkretnego ogłoszenia,
- formularz wprowadzania wiadomości z walidacją,
- zabezpieczenie dostępu do czatu tylko dla zalogowanych użytkowników.

Wiadomości zapisywane są w tabeli `wiadomosci` i odczytywane przez zapytania AJAX (w formacie JSON).

10. Bezpieczeństwo i walidacja

Projekt SklepOnline został wyposażony w podstawowe mechanizmy zabezpieczeń:

- Walidacja danych po stronie serwera i klienta.
- Ochrona przed SQL Injection przy pomocy `prepare()` i `bindParam()`.
- Ochrona przed XSS przy pomocy `htmlspecialchars()`.
- Bezpieczne logowanie i haszowanie haseł (`password_hash()`, `password_verify()`).
- Ograniczenie funkcji dostępnych tylko dla zalogowanych użytkowników.

Ponadto system rozróżnia użytkowników autoryzowanych od gości, zabezpieczając dostęp do formularzy czatu, dodawania ogłoszeń i zakupu.

11. Interfejs użytkownika

Wygląd aplikacji oparty jest na własnoręcznie napisanym CSS. Układ strony został zoptymalizowany pod kątem prostoty oraz intuicyjności:

- Jasny i ciemny motyw przełączany przez przycisk i zapamiętywany w localStorage.
- Karty ogłoszeń z podziałem na miniaturę i opis.
- Wspólny nagłówek (logo, przyciski logowania, wylogowania, przejścia do czatu i dodawania ogłoszeń).
- Spójna kolorystyka oparta na neutralnych barwach.

12. Testowanie

Testowanie aplikacji przeprowadzono lokalnie, z wykorzystaniem XAMPP. Testowano następujące scenariusze:

- Poprawna rejestracja i logowanie.
- Dodanie ogłoszenia i jego wyświetlanie.
- Filtracja ogłoszeń według kategorii.
- Symulacja płatności.
- Wymiana wiadomości między użytkownikami.

Wszystkie testy zakończyły się powodzeniem. System zachowuje się zgodnie z oczekiwaniami.

13. Możliwości rozwoju

Aplikacja stanowi dobrą podstawę do dalszej rozbudowy.

Możliwe ścieżki rozwoju to:

- Wdrożenie prawdziwego systemu płatności (np. PayPal).
- Wprowadzenie systemu ocen i komentarzy dla ogłoszeń.
- Rozbudowa systemu wiadomości o powiadomienia i załączniki.
- Rozszerzenie panelu użytkownika o historię zakupów.
- Responsywny design dopasowany do urządzeń mobilnych.

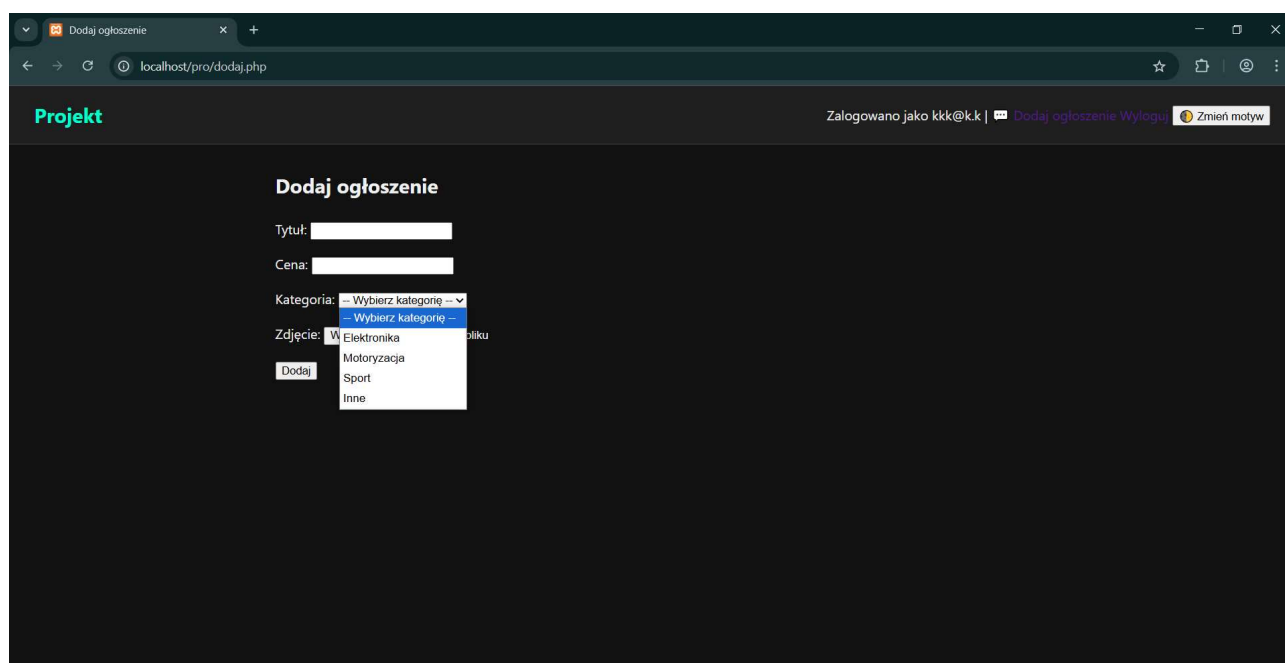
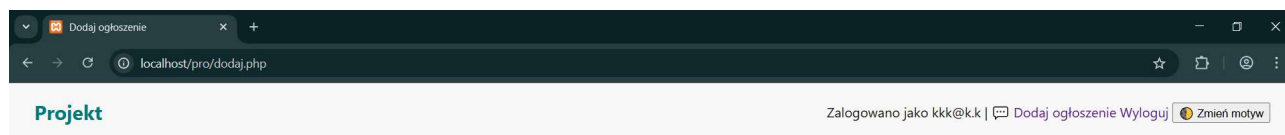
14. Podsumowanie

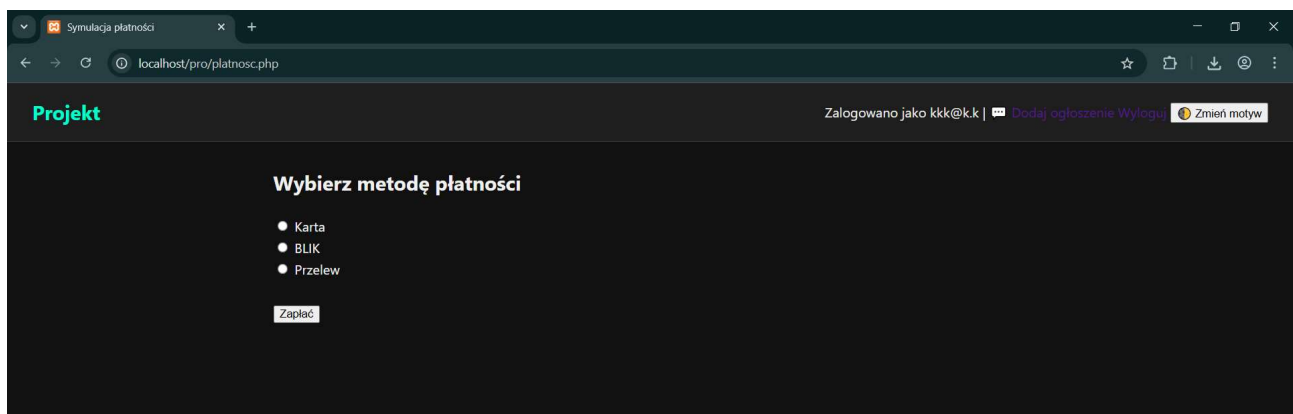
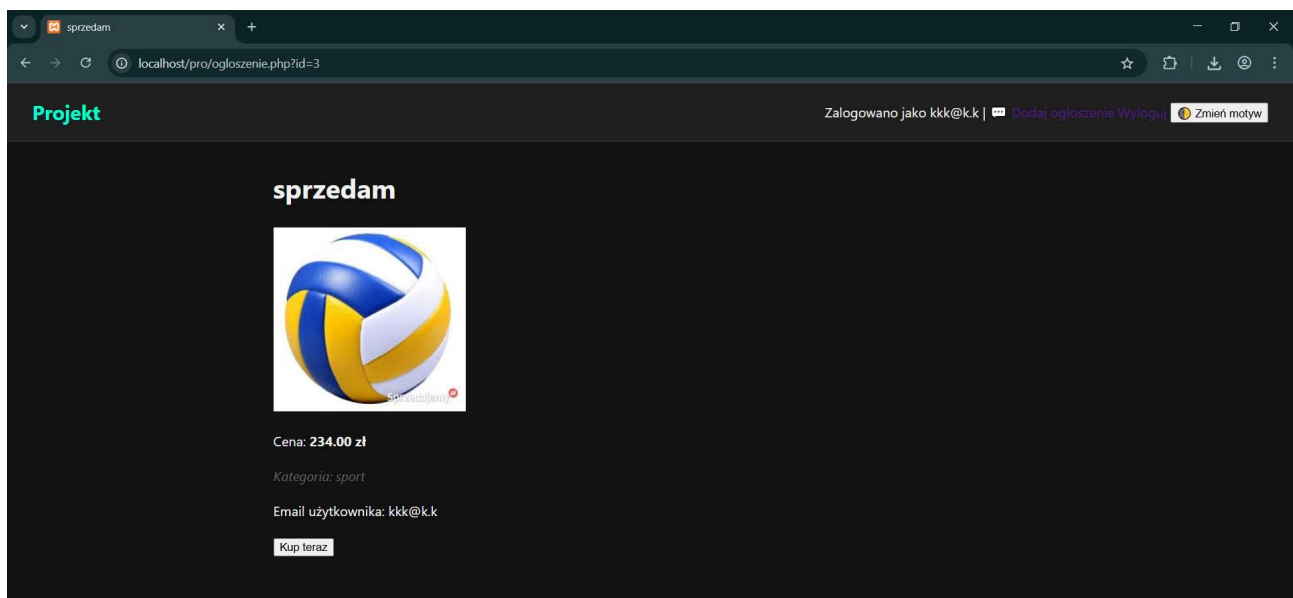
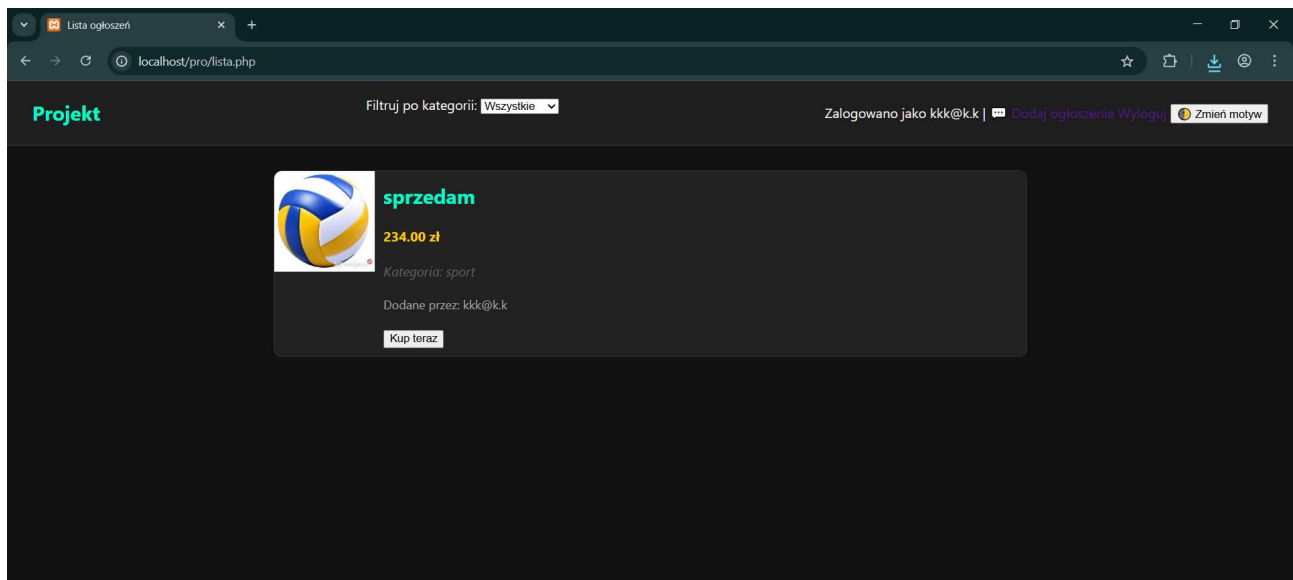
Projekt SklepOnline jest samodzielnie zaprojektowaną i zrealizowaną aplikacją ogłoszeniowo-sprzedażową. Została stworzona w całości w oparciu o czyste technologie bez frameworków.

Zastosowanie lokalnej bazy danych, dynamicznego czatu oraz symulowanego procesu zakupowego pozwala traktować aplikację jako pełnoprawny prototyp systemu e-commerce.

Kod źródłowy aplikacji, dokumentacja oraz baza danych zostały przygotowane w sposób umożliwiający łatwe uruchomienie i testowanie projektu lokalnie.

15.Screeny





Dziękujemy za zakup!

Płatność metodą „przelew” zakończona sukcesem. Jest to tylko atrapa systemu płatności i nie przetwarzam żadnych danych.

[Powrót do listy ogłoszeń](#)