

Sprawozdanie Megusta+
Projektowanie i programowanie
systemów internetowych I
prowadzący: mgr inż. Krzysztof Rewak

Grupa s2PAM2u Andrzej Czabajski, Dominik Kamiński, Jan Łabaj

7 czerwca 2024

Spis treści

1	Wstęp	2
2	Opis Funkcjonalny Systemu	3
3	Opis Technologiczny	8
4	Wyszczególnione Wdrożone Zagadnienia Kwalifikacyjne	9
5	Instrukcja Lokalnego i Zdalnego Uruchomienia Systemu	11
5.1	Lokalna Instalacja	11
5.1.1	Klonowanie Repozytorium	11
5.1.2	Konfiguracja Bazy Danych	11
5.1.3	Uruchomienie Serwera	11
5.1.4	Odwiedź Stronę	11
5.2	Zdalna Instalacja	12
5.2.1	Wdrażanie na Serwerze Zdalnym	12
5.2.2	Konfiguracja Bazy Danych	12
5.2.3	Konfiguracja Serwera Webowego	12
5.2.4	Konfiguracja DNS	12
5.2.5	Odwiedź Stronę	12
6	Wnioski Projektowe	13
6.1	Rekomendacje	13

Rozdział 1

Wstęp

W dzisiejszych czasach, kiedy coraz większą wagę przykładamy do zdrowego stylu życia i świadomego odżywiania, niezwykle istotne staje się monitorowanie i ocena spożywanych produktów. Aplikacja **Megusta** została stworzona z myślą o użytkownikach, którzy chcą aktywnie zarządzać swoją dietą, oceniać produkty spożywcze i podejmować lepsze decyzje żywieniowe.

Megusta to innowacyjne narzędzie, które umożliwia użytkownikom ocenianie produktów spożywczych na podstawie ich własnych preferencji smakowych, wartości odżywczych oraz innych kryteriów.



Rozdział 2

Opis Funkcjonalny Systemu

Megusta oferuje następujące główne funkcjonalności:

- **Wielojęzyczność:** Strona obsługuje dwa języki polski i angielski poprzez wykrycie preferencji językowych przeglądarki użytkownika i ustawiając odpowiednie ciasteczko.

```
<?php
if(isset($_COOKIE['lang']) && $_COOKIE['lang'] == 'pl') {
    setcookie("lang", "en");
} else if(isset($_COOKIE['lang']) && $_COOKIE['lang'] == 'en') {
    setcookie("lang", "pl");
} else {
    setcookie("lang", "", time() - 3600);
}

$url = explode( separator: "/", $_SERVER['HTTP_REFERER']);
header( header: "Location: ".$url[sizeof($url)-1]);
```

- **Routing:** System routingu kieruje żądania URL do odpowiednich modułów, umożliwiając dynamiczne generowanie treści.

Routing na tej stronie działa poprzez klasę Router, która zarządza ścieżkami URL i kieruje żądania do odpowiednich modułów w aplikacji. Oto kluczowe elementy działania routingu:

Inicjalizacja:

Kiedy użytkownik odwiedza stronę, index.php tworzy instancję klasy Router, przekazując jej obiekt bazy danych, bieżący URI oraz logger.

Konstruktor klasy Router ustawia globalną zmienną administratora i loguje żądanie użytkownika. Analiza URL:

Metoda `getParameters()` rozbija URI na segmenty i usuwa pierwsze dwa elementy, co pozwala na uzyskanie parametrów ścieżki. Ładowanie Modułu:

Metoda `loadModule()` tworzy instancję klasy `HTMLGenerator` i sprawdza, czy istnieje odpowiedni szablon w zależności od języka ustawionego w ciasteczkach. Jeśli szablon istnieje, ustawia go jako katalog szablonów. Jeśli użytkownik jest administratorem, ustawia szablon panelu administracyjnego. Generuje i wyświetla stronę za pomocą metody `generate()` klasy `HTMLGenerator`. Sprawdzenie Administratora:

Metoda `isAdmin()` sprawdza, czy ciasteczko użytkownika zawiera token administratora i zwraca odpowiednią wartość logiczną. Ustawienie Globalnego Administratora:

Metoda `setAdminGlobal()` wykonuje zapytanie do bazy danych, aby ustawić token administratora jako globalną zmienną. Dzięki tym krokom, klasa Router efektywnie zarządza routingiem na stronie, kierując użytkowników do odpowiednich szablonów i modułów na podstawie URL, języka oraz uprawnień użytkownika.

- **Panel Administracyjny:** Umożliwia zarządzanie treścią, która docelowo znajdzie się na stronie. Tylko Administrator może dodawać nowe produkty i opisy.

Dodaj Produkt

Zdjęcie produktu

Wybierz plik

Nie wybrano pliku

Nazwa produktu

Wpisz nazwę produktu

Opis produktu

Opisz produkt

Cena produktu


Podaj cenę Produktu

Dodaj produkt

- **Logowanie Aktywności:** Logowanie informacji i błędów w celu monitorowania i debugowania aplikacji.



- **Interfejs Użytkownika:** Responsywny interfejs użytkownika stylizowany za pomocą Tailwind CSS, zapewniający estetyczny i spójny wygląd.



mleko UHT 3.2%
super mleko pasteryzowane aż 3,2 % tłuszczu
Cena: 2 zł
Ocena: 0 - ☆☆☆☆☆

Ocena:

Komentarz:

Wyślij

Rozdział 3

Opis Technologiczny

Projekt Megusta wykorzystuje następujące technologie:

- **PHP:** Główny język programowania używany do tworzenia logiki serwera i dynamicznego generowania treści.
- **MySQL:** System zarządzania bazą danych, który przechowuje dane aplikacji, takie jak użytkownicy i treści.
- **Tailwind CSS:** Framework CSS typu utility-first, używany do stylizacji elementów HTML w aplikacji.
- **Composer:** Menedżer zależności PHP, który automatycznie instaluje i zarządza zewnętrznymi bibliotekami i komponentami.
- **Apache:** Serwer webowe używany do hostowania aplikacji.

Rozdział 4

Wyszczególnione Wdrożone Zagadnienia Kwalifikacyjne

Projekt Megusta obejmuje wdrożenie kilku istotnych zagadnień kwalifikacyjnych:

- **Zarządzanie Sesjami i Autoryzacja:** Obsługa sesji użytkowników, w tym logowanie, ciasteczka językowe oraz autoryzacja dostępu do panelu administracyjnego.
- **Bezpieczeństwo Aplikacji:** Zabezpieczenia przed nieautoryzowanym dostępem do panelu administracyjnego oraz sanitacja danych wejściowych do bazy danych w celu zapobiegania atakom SQL injection.
- **Logowanie i Monitorowanie:** Implementacja klasy `MyLogger` do logowania aktywności użytkowników i błędów systemowych.
- **Dynamiczne Generowanie Treści:** Użycie systemu szablonów i klasy `HTMLGenerator` do dynamicznego generowania stron na podstawie parametrów URL i danych z bazy.
- **Wielojęzyczność:** Automatyczne wykrywanie preferencji językowych przeglądarki i dostosowywanie treści strony do wybranego języka.

#	Zagadnienie	Opis	Wdrożenie
1	framework MVC	wykorzystanie frameworka na backen- dzie	-
2	framework CSS	wykorzystanie frameworka na fronten- dzie	+
3	baza danych	dołączenie do projektu bazy danych	+
4	cache	dołączenie do projektu systemu cache	-
5	dependency manager	dołączenie do projektu systemu zarzą- dzania zależnościami	+
6	HTML	szkielet aplikacji internetowej	+
7	CSS	ostylowanie aplikacji internetowej	+
8	JavaScript	uinteraktywnienie aplikacji interneto- wej	+
9	routing	wykorzystany routing i tzw. pretty URLs	+
10	ORM	wykorzystane mapowanie obiektowo- relacyjne	+
11	uwierzytelnianie	zaimplementowane mechanizmy uwie- rzytelnienia	+
12	lokalizacja	możliwość przełączania języka aplikacji	+
13	mailing	wysyłanie mejli z aplikacji	+
14	formularze	przesyłanie danych do aplikacji przez formularze	+
15	asynchroniczne inte- rakcje	zaimplementowane asynchroniczne in- terakcje z serwerem	+
16	konsumpcja API	wykorzystanie zewnętrznego API	+
17	publikacja API	wystawienie własnego API	-
18	RWD	responsywny frontend	+
19	logger	logowanie akcji w systemie	+
20	deployment	wdrożenie aplikacji internetowej	+

Rozdział 5

Instrukcja Lokalnego i Zdalnego Uruchomienia Systemu

5.1 Lokalna Instalacja

5.1.1 Klonowanie Repozytorium

Sklonuj repozytorium na lokalny serwer i zainstaluj zależności za pomocą Composer.

```
git clone https://github.com/yourusername/megusta.git
cd megusta
composer install
```

5.1.2 Konfiguracja Bazy Danych

Skonfiguruj połączenie z bazą danych w pliku DB.php.

```
$this->db = new mysqli("host", "user", "password", "database");
```

5.1.3 Uruchomienie Serwera

Skonfiguruj serwer Apache. Przenieś wszystkie pliki do katalogu htdocs. Zaimportuj bazę danych.

5.1.4 Odwiedź Stronę

Otwórz przeglądarkę i przejdź pod adres <http://localhost/index.php>.

5.2 Zdalna Instalacja

5.2.1 Wdrażanie na Serwerze Zdalnym

Zaloguj się na serwer zdalny i sklonuj repozytorium na serwer. W przypadku korzystania ze środowiska IDE PHP Storm możesz to zrobić poprzez funkcję Deployment.

```
git clone https://github.com/yourusername/megusta.git
cd megusta
composer install
```

5.2.2 Konfiguracja Bazy Danych

Skonfiguruj połączenie z bazą danych w pliku DB.php.

```
$this->db = new mysqli("host", "user", "password", "database");
```

5.2.3 Konfiguracja Serwera Webowego

Skonfiguruj serwer Apache, aby wskazywał na katalog projektu. Upewnij się, że moduły PHP są włączone (np. mod_php w Apache).

5.2.4 Konfiguracja DNS

Skonfiguruj DNS, aby domena wskazywała na adres IP serwera zdalnego.

5.2.5 Odwiedź Stronę

Otwórz przeglądarkę i przejdź pod adres <https://host358482.xce.pl/>.

Rozdział 6

Wnioski Projektowe

Projekt **Megusta** jest przykładem aplikacji webowej, która łączy różne technologie i rozwiązania w celu stworzenia funkcjonalnego, responsywnego i bezpiecznego systemu zarządzania treścią. Wdrożenie wielojęzyczności, dynamicznego generowania treści oraz systemu logowania aktywności pozwala na elastyczne zarządzanie treścią i użytkownikami. Stylizacja za pomocą Tailwind CSS zapewnia spójny i estetyczny wygląd interfejsu użytkownika.

6.1 Rekomendacje

- Dalsza optymalizacja kodu pod kątem wydajności i bezpieczeństwa. Uwzględnienie w rozwoju projektu wykorzystania frameworka - Laravel.
- Rozszerzenie funkcjonalności panelu administracyjnego o dodatkowe moduły zarządzania.
- Regularne aktualizacje zależności i komponentów w celu zapewnienia zgodności z najnowszymi standardami.