WYŻSZA SZKOŁA BANKOWA W GDAŃSKU

WYDZIAŁ FINANSÓW I ZARZĄDZANIA

Dawid Zamorowski

nr albumu 42449

**PORÓWNANIE WYDAJNOŚCI APLIKACJI WEBOWEJ W TECHNOLOGII PHP I JAVASCRIPT**

Praca magisterska

na kierunku Informatyka

Praca napisana pod kierunkiem

prof. dr. inż. Mariusza Mola

Gdańsk 2023

**Streszczenie**

        Temat: Porównanie wydajności aplikacji webowej w technologii PHP i JavaScript

        Słowa kluczowe: PHP, JavaScript, aplikacja webowa, wydajność, porównanie

        Celem pracy jest przetestowanie i porównanie wydajności różnych narzędzi dostępnych dla PHP i JavaScript w celu poprawy wydajności aplikacji webowej.

**Abstract**

        Topic: Comparison of performance of web application in PHP and JavaScript technology

Keywords: PHP, JavaScript, web application, performance, comparison

The aim of the study is to test and compare the performance of different tools available for PHP and JavaScript, in order to improve the performance of web applications.

**Oświadczenie o samodzielnym wykonaniu pracy**

        Stwierdzam, że przedstawiona praca dyplomowa została napisana przeze mnie samodzielnie i nie zawiera treści uzyskany w sposób niezgodny z obowiązującymi przepisami. Oświadczam ponad to, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną.

Podpis:..............................................

**Spis** **treści**

**Streszczenie**

**Abstract**

**Oświadczenie o samodzielnym wykonaniu pracy**

**Wstęp**

1. **Technologie i narzędzia**
   1. **Język PHP**
   2. **Język JavaScript**
   3. **Baza danych**
   4. **Narzędzia do testowania i debugowania**
      1. **Uruchamianie narzędzia**
      2. **Elements**
      3. **Inspect**
      4. **Sekcja CSS**
      5. **Edycja kodu i CSS**
      6. **Analiza wersji mobilnej**
      7. **Console**
      8. **Analiza sieciowa**
      9. **Audyt**
2. **Implementacja aplikacji w języku PHP**
   1. **Baza danych**
   2. **Logowanie**
   3. **Wylogowanie**
   4. **Rejestracja**
   5. **Aktualny użytkownik**
   6. **Usunięcie danych**
   7. **Aktualizacja danych**
   8. **Dodanie elementów**
3. **Implementacja aplikacji w języku JavaScript**
   1. **Baza danych**
   2. **Logowanie**
   3. **Wylogowanie**
   4. **Rejestracja**
   5. **Aktualny użytkownik**
   6. **Usunięcie danych**
   7. **Aktualizacja danych**
   8. **Dodanie elementu**
4. **Testy wydajnościowe**
   1. **Metodologia testowania**
   2. **Narzędzia do testowania wydajności**
   3. **Wyniki testów i ich analiza**
5. **Podsumowanie i wnioski**
   1. **Porównanie wydajności aplikacji w PHP i JavaScript**
   2. **Ocena projektu aplikacji webowej do serwisowania samochodów osobowych**
   3. **Propozycje usprawnień i rozwoju aplikacji**

**Zawartość płyty**

**Bibliografia i netografia**

**Wstęp**

W obecnych czasach rozwój technologii internetowych i aplikacji webowych jest nieunikniony. Wiele dziedzin życia korzysta z takiej formy dostępu do informacji i przetwarzania danych, a jednym z nich jest branża motoryzacyjna. Serwisowanie samochodów osobowych to jedna z najważniejszych gałęzi tej branży, a aplikacje webowe mogą pomóc w usprawnieniu tego procesu.

Celem niniejszej pracy magisterskiej jest stworzenie aplikacji webowej do serwisowania samochodów osobowych oraz porównanie wydajności tej aplikacji napisanej w języku PHP oraz JavaScript. Obejmuje ona wiele etapów, w tym implementację aplikacji oraz testy wydajnościowe, które pozwolą na dokładne porównanie obu wersji aplikacji.

W pierwszej części pracy zostaną przedstawione narzędzia i technologie, które zostały wykorzystane w projekcie aplikacji webowej. Omówione zostaną języki programowania, takie jak PHP i JavaScript, a także narzędzia do testowania i debugowania aplikacji.

Następnie zostaną przedstawione szczegóły implementacji aplikacji w języku PHP oraz w języku JavaScript. Opisane zostaną poszczególne moduły aplikacji oraz sposób ich implementacji. Będzie to również okazja do przedstawienia sposobu organizacji kodu w projekcie, a także zagadnień związanych z bezpieczeństwem i optymalizacją kodu.

Kolejnym ważnym etapem będzie przeprowadzenie testów wydajnościowych aplikacji napisanej w języku PHP oraz JavaScript. Testy te pozwolą na dokładne porównanie wydajności obu wersji aplikacji, a także określenie, która z nich działa szybciej i efektywniej. W celu przeprowadzenia tych testów zostaną wykorzystane różne narzędzia i metody, które pozwolą na uzyskanie wiarygodnych wyników.

Ostatecznie, w części podsumowującej zostaną przedstawione wnioski wynikające z porównania wydajności aplikacji napisanych w językach PHP i JavaScript, a także ogólna ocena projektu aplikacji webowej do serwisowania samochodów osobowych. Będzie to również okazja do przedstawienia ewentualnych usprawnień, które mogłyby zostać wprowadzone w celu poprawy działania aplikacji.

Celem pracy magisterskiej jest przedstawienie, w jaki sposób aplikacje webowe mogą usprawnić proces serwisowania samochodów osobowych, a także dokładne porównanie wydajności aplikacji napisanych w językach PHP i JavaScript. W ten sposób można zrozumieć, jakie narzędzia i technologie są najlepsze do stworzenia aplikacji webowych o wysokiej wydajności, a także jakie wyzwania i problemy mogą pojawić się podczas tworzenia aplikacji.

1. **Technologie i narzędzia**

Technologie webowe odgrywają coraz większą rolę w dzisiejszym świecie. Aplikacje internetowe stały się nieodłącznym elementem biznesu, edukacji, rozrywki i codziennego życia. W tym rozdziale przedstawię języki programowania i narzędzia, które wykorzystałem do porównania wydajności aplikacji webowej napisanej w PHP i JavaScript.

* 1. **Język PHP**

Język PHP (Hypertext Preprocessor) to popularny język skryptowy, często wykorzystywany do tworzenia stron internetowych i aplikacji webowych. Jego główną zaletą jest to, że jest łatwy do nauczenia i używania, a także dobrze integruje się z bazami danych i serwerami HTTP.

W PHP, kod jest wykonywany po stronie serwera, co oznacza, że cały kod PHP jest przetwarzany na serwerze, a wynik jest wysyłany do przeglądarki internetowej klienta. To umożliwia tworzenie dynamicznych stron internetowych, w których treść zmienia się w zależności od działań użytkownika.

Język PHP posiada wiele wbudowanych funkcji, które ułatwiają pracę programistom. Możliwe jest także łatwe łączenie się z bazami danych, takimi jak MySQL, Oracle czy PostgreSQL, co umożliwia przechowywanie danych użytkowników i innych informacji związanych z aplikacją.

Jedną z wad PHP jest to, że nie jest to język, który został zaprojektowany do budowania dużych aplikacji. Zamiast tego, PHP jest często używany do tworzenia małych skryptów i aplikacji webowych o niskiej i średniej złożoności. W przypadku bardziej skomplikowanych aplikacji, mogą pojawić się problemy z wydajnością i skalowalnością.

W celu poprawy wydajności aplikacji napisanych w PHP, istnieją różne techniki, takie jak cacheowanie, optymalizacja kodu i korzystanie z frameworków PHP, które oferują wiele wbudowanych narzędzi i bibliotek. Warto również pamiętać, że PHP jest ciągle rozwijany i ulepszany, dzięki czemu możliwe jest tworzenie coraz bardziej wydajnych aplikacji.

* 1. **Język JavaScript**

Język JavaScript to inny popularny język programowania, którego głównym zastosowaniem jest tworzenie dynamicznych interaktywnych stron internetowych. W przeciwieństwie do PHP, JavaScript działa po stronie klienta, co oznacza, że kod jest wykonywany w przeglądarce internetowej użytkownika, a nie na serwerze.

JavaScript umożliwia tworzenie efektów wizualnych, takich jak animacje, przejścia i inne interaktywne elementy na stronach internetowych. Jest to możliwe dzięki DOM (Document Object Model), który reprezentuje strukturę strony internetowej i umożliwia interakcję z jej elementami.

Język JavaScript ma wiele wbudowanych funkcji i bibliotek, które ułatwiają pracę programistom. Najpopularniejszą biblioteką JavaScript jest jQuery, która oferuje wiele gotowych funkcji, takich jak manipulacja elementami DOM, obsługa zdarzeń czy animacje.

JavaScript również integruje się z bazami danych, ale w przeciwieństwie do PHP, wymaga do tego API (Application Programming Interface) takie jak np. Firebase. Firebase to platforma Firebase umożliwiająca szybkie i łatwe tworzenie aplikacji internetowych bez konieczności posiadania własnego serwera.

Jedną z wad JavaScript jest to, że jego wydajność zależy od wydajności przeglądarki internetowej i sprzętu użytkownika. Ponadto, różnice w implementacji JavaScriptu w różnych przeglądarkach mogą prowadzić do problemów z kompatybilnością.

W celu poprawy wydajności aplikacji napisanych w JavaScript, istnieją różne techniki, takie jak cacheowanie, optymalizacja kodu i korzystanie z nowych funkcjonalności języka. Warto również pamiętać, że JavaScript jest ciągle rozwijany i ulepszany, dzięki czemu możliwe jest tworzenie coraz bardziej wydajnych aplikacji.

* 1. **Baza danych**

Baza danych to kluczowy element większości aplikacji webowych, który umożliwia przechowywanie i zarządzanie informacjami w sposób efektywny i bezpieczny. W niniejszej pracy wykorzystano bazę danych MySQL, która jest jednym z najpopularniejszych systemów zarządzania bazami danych.

MySQL jest relacyjną bazą danych, co oznacza, że dane są przechowywane w tabelach i relacjach między nimi. MySQL oferuje wiele funkcji, takich jak obsługa transakcji, klucze obce i indeksowanie, które pozwalają na efektywne zarządzanie i wyszukiwanie danych.

W przypadku aplikacji webowych, baza danych MySQL jest zwykle wykorzystywana do przechowywania informacji o użytkownikach, takich jak nazwiska, adresy e-mail, loginy i hasła. Baza danych może również przechowywać informacje o produktach, zamówieniach i innych danych biznesowych.

Ważnym aspektem bazy danych jest jej wydajność, zwłaszcza w przypadku aplikacji webowych z dużym ruchem. W celu poprawy wydajności, można stosować różne techniki, takie jak indeksowanie, partycjonowanie i cacheowanie.

W niniejszej pracy wykorzystano lokalną bazę danych MySQL, postawioną na XAMPP przy użyciu phpMyAdmin. XAMPP to pakiet oprogramowania służący do tworzenia i zarządzania serwerami WWW, w tym Apache, MySQL i PHP.

Podsumowując, baza danych MySQL jest kluczowym elementem aplikacji webowych i umożliwia efektywne zarządzanie informacjami. W celu poprawy wydajności, należy stosować różne techniki i narzędzia.

* 1. **Narzędzia do testowania i debugowania**

Chrome DevTools to zbiór narzędzi wbudowanych w przeglądarkę, które umożliwiają analizę i edycję stron internetowych. Dzięki temu narzędziu można wykryć i naprawić występujące problemy oraz zobaczyć implementacje wybranego elementu, modułu lub sekcji. Podobne narzędzia dostępne są także w innych przeglądarkach, takich jak Firefox, Safari czy Opera.

* + 1. **Uruchamianie narzędzia**

Aby otworzyć DevTools, istnieje kilka sposobów. Można kliknąć prawym przyciskiem myszy na stronie i wybrać opcję "Zbadaj" lub "Inspect". Można również skorzystać ze skrótu klawiszowego Ctrl + Shift + C (Windows, Linux, Chrome OS) lub Command + Option + C (Mac). Inną opcją jest wybranie "Narzędzia dla deweloperów" z menu przeglądarki.

* + 1. **Elements**

Warto zaznaczyć, że DevTools pozwala nie tylko na analizę kodu strony, ale również na jego edycję. Możemy bezpośrednio w DevTools zmieniać atrybuty, dodawać nowe elementy czy modyfikować już istniejące. Dzięki temu możemy szybko sprawdzić, jakie efekty przyniesie dana zmiana bez konieczności wprowadzania jej w kodzie źródłowym. Jest to szczególnie przydatne w przypadku testowania różnych wariantów stylów czy układów elementów.

DevTools oferuje również narzędzia pozwalające na debugowanie kodu JavaScript. Możemy w łatwy sposób zlokalizować błędy w skrypcie i monitorować zmienne, co znacznie ułatwia proces debugowania. Możemy też dodać breakpointy, czyli punkty zatrzymania, w określonych miejscach kodu, co pozwala na analizę stanu programu w danym momencie.

Oprócz tego DevTools udostępnia wiele innych narzędzi, np: Performance, które pozwala na analizę wydajności strony, Network, dzięki któremu możemy śledzić żądania i odpowiedzi serwera oraz zidentyfikować przyczyny spowolnień w ładowaniu strony.

Podsumowując, Chrome DevTools to niezwykle przydatne narzędzie dla każdego web developera. Pozwala na szybką i efektywną analizę oraz edycję kodu strony, debugowanie kodu JavaScript oraz analizę wydajności strony.

* + 1. **Inspect**

Jest to bardzo przydatne narzędzie, które umożliwia szybkie zlokalizowanie kodu i stylów odpowiadających konkretnemu elementowi na stronie. W ten sposób możemy szybko i precyzyjnie wprowadzać zmiany w wyglądzie i funkcjonalności strony. Warto także zaznaczyć, że DevTools pozwala na dokonywanie zmian w kodzie strony w czasie rzeczywistym, co może być bardzo pomocne w procesie debugowania lub wprowadzania drobnych poprawek.

* + 1. **Sekcja** **CSS**

Oprogramowanie pozwala na przeglądanie oraz dynamiczną edycję stylów, które opisują wygląd i zachowanie analizowanego elementu na stronie internetowej. Interesującą funkcjonalnością jest możliwość wymuszenia pewnych zachowań elementu, takich jak: hover, active, visited czy focus, dla których można określić inne style lub zachowania.

* + 1. **Edycja** **kodu** **i** **CSS**

DevTools umożliwia dynamiczną edycję kodu i CSS strony, jednak po odświeżeniu strony wszystkie zmiany zostaną zresetowane. Niemniej jednak, jest to fajny sposób na dodawanie, usuwanie, ukrywanie czy edycję elementów na stronie (znaczników, tekstu, linków, kolorów, fontów, rozmiarów itd.). Jest to przydatne, gdy chcemy zmienić lub dodać coś na stronie, lub przygotować screenshot. Taka możliwość pozwala nam przetestować wprowadzane zmiany bez konsekwencji. Później wystarczy wprowadzić zmiany w plikach strony lub w systemie CMS. W przypadku edycji CSS przydatna jest informacja o pliku i linii kodu, w którym znajduje się dana reguła stylów.

Wyszukiwarka - Nie wyobrażam sobie pracy bez możliwości przeszukiwania kodu w poszukiwaniu istotnych elementów z punktu widzenia pozycjonowania. Dzięki wyszukiwarce możemy wyszukiwać:

* dowolny ciąg znaków,
* kwerendy CSS np: .nazwa klasy, czy #identyfikator i określić ile występuje danych elementów, bądź też sprawdzić jakie inne elementy,
* korzystają ze stylów analizowanej klasy CSS,
* konkretne znaczniki, np: //h1, //p, //img, //a, itp.
  + 1. **Analiza** **wersji** **mobilnej**

Te narzędzie umożliwia analizowanie responsywności strony oraz wyświetlania jej elementów na różnych urządzeniach przenośnych. Dzięki niemu możemy sprawdzić, czy strona działa poprawnie na różnych rozdzielczościach ekranu. Jest to istotne z punktu widzenia SEO, ponieważ strony przyjazne urządzeniom mobilnym przyciągają więcej ruchu. Narzędzie umożliwia wybór różnych urządzeń, skonfigurowanie własnych lub wybranie opcji "responsive", która pozwala na dowolną zmianę szerokości i wysokości okna przeglądarki. Możemy również ręcznie wprowadzać wartości i zmieniać rotację ekranu. Kolejną przydatną funkcją jest możliwość symulowania opóźnień wczytywania strony, które wpływają na jej wydajność (no throttling, mid-tier mobile, low-tier mobile i offline).

* + 1. **Console**

To zakładka, z której korzystam głównie podczas audytowania stron internetowych, szczególnie gdy występują jakieś błędy lub ostrzeżenia. Czasami trudne problemy wymagają konsultacji z zespołem Web Development, aby upewnić się, że naprawy nie wpłyną negatywnie na działanie strony, SEO lub UX.

* + 1. **Network**

Zakładka "Network" umożliwia dokładne badanie ruchu sieciowego między serwerem a przeglądarką użytkownika. Dzięki niej możemy łatwo sprawdzić:

* jakie zasoby otrzymujemy od serwera,
* metodę przesyłania pakietów (np: http/1, http/2),
* status odpowiedzi na zapytanie,
* typ i rozmiar przesyłanego pliku,
* czas odpowiedzi i dostarczenia zasobu,
* itd.

Możemy zidentyfikować kilka elementów, które przyczyniają się do dłuższego czasu ładowania strony.

Przy podstawowej analizie strony najpierw zaznaczam pole wyboru "cache", aby zasoby ładowały się z serwera, a nie z pamięci podręcznej przeglądarki po odświeżeniu strony. W zakładce "Status" sprawdzam, czy wszystkie zasoby mają kod 200. W zakładce "Protocol" określam, jakie metody są wykorzystywane do wysyłania zasobów z serwera, ponieważ wpływa to na szybkość ładowania strony. W sekcji "Size" analizuję wagę przesyłanych plików, a w kolumnie "Time" zwracam uwagę na czas ładowania.

* + 1. **Lighthouse**

Krótko mówiąc, LightHouse to rozszerzenie dla przeglądarki Google Chrome, które kompleksowo przeprowadza audyt strony internetowej, zwracając szczególną uwagę na jakość, wydajność, SEO i dostosowanie do potrzeb użytkowników. Wynik testu podzielony jest na 5 sekcji: Wydajność, Dostępność, Najlepsze praktyki, SEO i Progressive Web App, dla każdej z nich przypisując osobną wartość wyniku w przedziale od 0 do 100 punktów. Im wyższy wynik, tym lepiej. Warto również zauważyć, że narzędzie Google PageSpeed Insights, które określa szybkość ładowania strony internetowej, opiera się na analizie LightHouse.

1. **Implementacja aplikacji w języku PHP**

W niniejszym rozdziale przedstawiona jest implementacja aplikacji webowej po stronie klienta oraz serwera. Klient, korzystając z internetu, przesyła zapytanie do serwera (bazy danych), aby otrzymać informacje o samochodzie. W tym celu wykorzystano PHP do stworzenia warstwy aplikacji, a MySQL jako system bazy danych



*Diagram : Diagram obrazujący implementacje serwera w kodzie PHP*

* 1. **Baza danych**



*Screenshot : Kod połączenia z bazą MySQL w kodzie PHP*

W powyższym kodzie zostało utworzone połączenie z bazą danych serwera, wykorzystując język PHP. Utworzony został plik config.php w celu stworzenia połączenia między aplikacją webową, a bazą danych. Dzięki temu, korzystając z jakichkolwiek zapytań do bazy, czy też wysyłania elementów do niej, możliwe było stworzenie prostej zmiennej związaną z połączeniem do bazy. To pozwoliło mi na łatwe zarządzanie połączeniem do bazy.

* 1. **Logowanie**



*Screenshot : Kod logowania się w kodzie PHP*

Powyższy kod umożliwia pobranie przez użytkownika danych z konkretnego pola, a następnie porównanie ich elementów z danymi znajdującymi się w bazie danych. Jeśli elementy będą do siebie pasować, użytkownik zostanie zalogowany do aplikacji. W przypadku niepoprawnego logowania, klient otrzyma informację o błędnym loginie lub haśle.

* 1. **Wylogowanie**



*Screenshot : Kod wylogowania się w kodzie PHP*

Podany kod pozwala na wylogowanie się z bieżącej sesji, w której użytkownik aktualnie przebywa, a jednocześnie umożliwia mu opuszczenie swojego konta.

* 1. **Rejestracja**



*Screenshot : Kod rejestracji w kodzie PHP*

Korzystając z rejestracji, można wprowadzać wszystkie dane, dodając nowego użytkownika do bazy danych. Oczywiście należy sprawdzić, czy wprowadzone już nazwy nie są przypadkiem zajęte

* 1. **Aktualny użytkownik**

****

*Screenshot : Kod aktualnie zalogowanego użytkownika w kodzie PHP*

Można wykorzystać powyższy kod do wyświetlenia danych takich jak imię i nazwisko, wprowadzone podczas rejestracji, oraz nazwy użytkownika, który jest obecnie zalogowany.

* 1. **Usunięcie danych**

****

*Screenshot : Kod usunięcia samochodu w kodzie PHP*

Powyższy kod umożliwia usuwanie danych z bazy, w tym przypadku dotyczy to samochodu osobowego. Przed usunięciem system poprosi o potwierdzenie wykonania tej czynności. Po zatwierdzeniu, użytkownik straci wszystkie informacje o serwisowaniach wykonanych na tym samochodzie.

* 1. **Aktualizacja danych**

****

*Screenshot : Kod aktualizacji przebiegu w kodzie PHP*

Opisany kod służy do weryfikacji wprowadzonych danych oraz ich porównania z wartością (przebiegiem) przechowywaną w bazie danych. Jeśli użytkownik poda wartość niższą lub równą aktualnemu przebiegowi pojazdu, to przebieg ten nie zostanie zaktualizowany.

* 1. **Dodanie elementów**



*Screenshot : Kod dodania serwisowania w kodzie PHP*



*Screenshot : Kod dodania samochodu w kodzie PHP*

Oba powyższe kody służą do dodawania nowych elementów do bazy danych. Pierwszy kod umożliwia wprowadzanie notatek serwisowych dla określonego samochodu i wyświetlanie ich w postaci listy. Z kolei drugi kod służy do dodawania nowego pojazdu, który zostanie przypisany do aktualnie zalogowanego profilu i będzie wyświetlany w postaci listy.

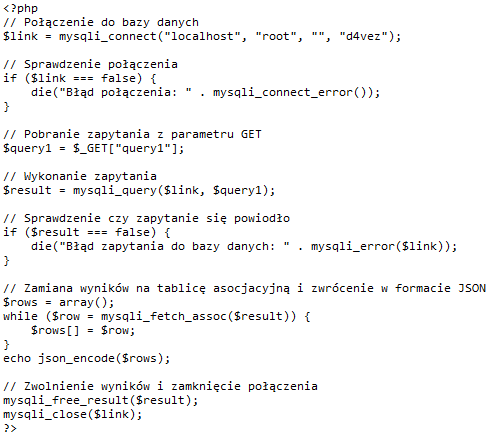
1. **Implementacja aplikacji w języku JavaScript**

W niniejszym rozdziale zostanie przedstawione omówienie implementacji kodu napisanego w języku JavaScript, którego zastosowanie dotyczy zarówno strony klienta jak i serwera w formie aplikacji internetowej. W opisywanym rozwiązaniu wykorzystano technologie i narzędzia opisane w poprzednich rozdziałach. Klient jest tutaj pojęciem odnoszącym się do aplikacji internetowej, która za pośrednictwem Internetu wysyła zapytania do serwera (w tym przypadku serwera z bazą danych), aby otrzymać informacje na temat samochodu. Do implementacji warstwy aplikacji wykorzystano język JavaScript, do wysyłania zapytań do bazy danych wykorzystano PHP, a jako system bazodanowy zastosowano MySQL.



*Diagram : Diagram obrazujący implementacje serwera w kodzie JavaScript*

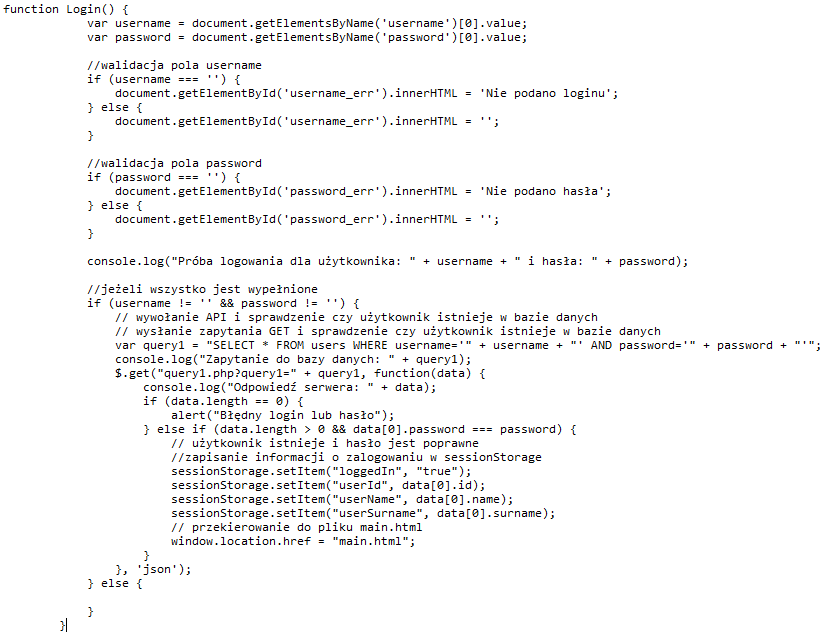
* 1. **Baza danych**



*Screenshot : Kod połączenia z bazą MySQL w kodzie JavaScript*

W przedstawionym kodzie zawarto połączenie z bazą danych serwera przy użyciu języka PHP. Utworzone zostały pliki query1.php oraz query2.php, które umożliwiły stworzenie połączenia między aplikacją internetową a bazą danych. Dzięki temu udało mi się stworzyć prostą zmienną związaną z połączeniem do bazy, co okazało się bardzo wartościowe przy korzystaniu z zapytań i wysyłaniu elementów do bazy danych.

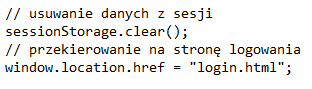
* 1. **Logowanie**



*Screenshot : Kod logowania się w kodzie JavaScript*

Ten kod pobiera informacje, które użytkownik wprowadza w określone pola, a następnie porównuje je z elementami znajdującymi się w bazie danych. Jeśli wprowadzone informacje są zgodne z danymi w bazie, użytkownik zostanie zalogowany do aplikacji. W przypadku nieudanego logowania, klient zostanie poinformowany o błędnej nazwie użytkownika lub haśle.

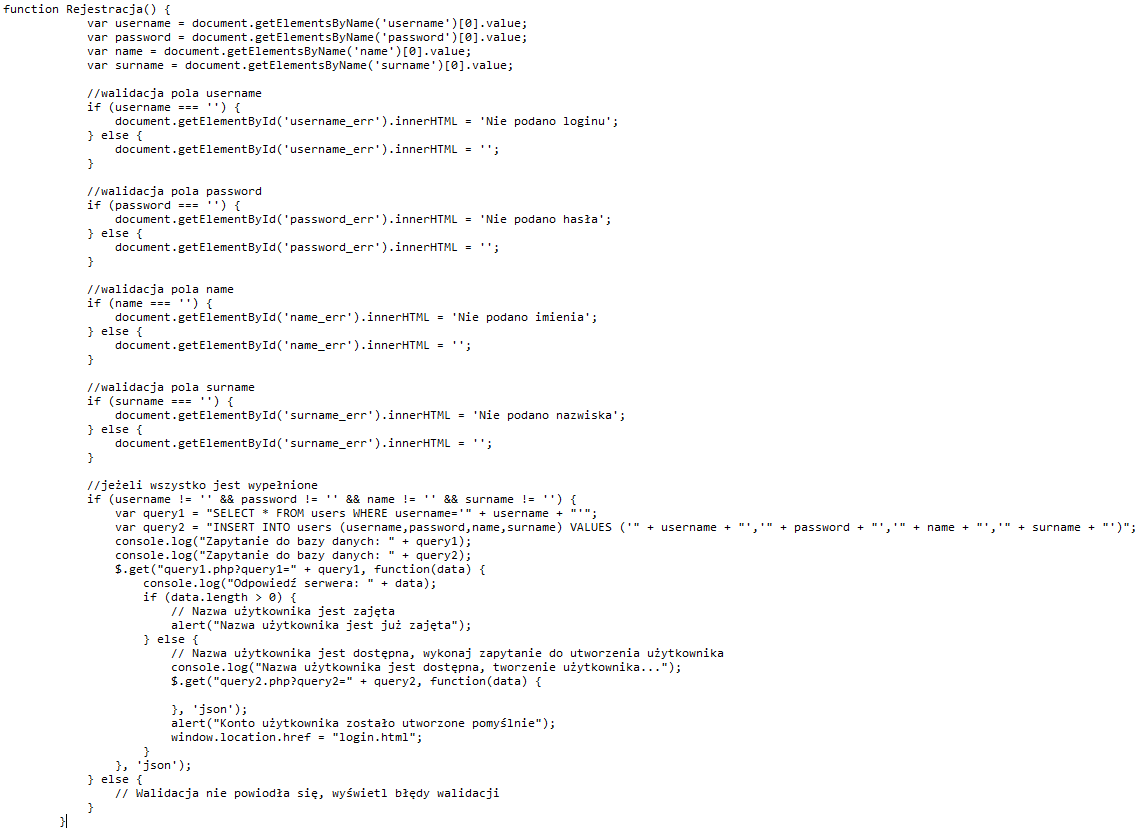
* 1. **Wylogowanie**



*Screenshot : Kod wylogowania się w kodzie JavaScript*

Ten kod umożliwia użytkownikowi wylogowanie się z aktualnej sesji, w której obecnie jest zalogowany, co pozwala mu na opuszczenie swojego konta.

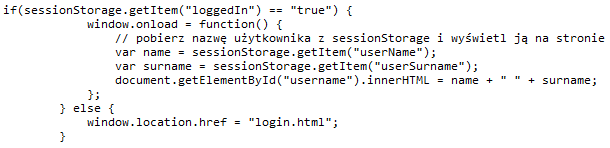
* 1. **Rejestracja**



*Screenshot : Kod rejestracji w kodzie JavaScript*

Kod rejestracji umożliwia dodanie nowego użytkownika do bazy danych po podaniu wszystkich wymaganych informacji. Jest również zaprogramowany, aby upewnić się, czy wprowadzona nazwa użytkownika nie jest już zajęta przez innego użytkownika.

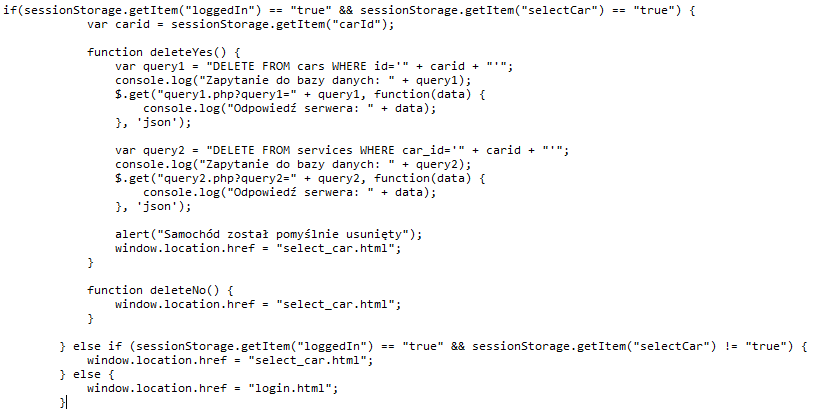
* 1. **Aktualny użytkownik**



*Screenshot : Kod aktualnie zalogowanego użytkownika w kodzie JavaScript*

Ten kod umożliwia użytkownikowi wylogowanie się z aktualnej sesji, w której obecnie jest zalogowany, co pozwala mu na opuszczenie swojego konta.

* 1. **Usunięcie danych**



*Screenshot : Kod usunięcia samochodu w kodzie JavaScript*

Przedstawiony powyżej kod pozwala na usunięcie danych z bazy danych, w tym przypadku dotyczących samochodu osobowego. Przy próbie usunięcia użytkownik zostanie poproszony o potwierdzenie tej czynności, aby upewnić się, że jest pewien swojej decyzji. Akceptując usunięcie samochodu, użytkownik straci również wszelkie informacje dotyczące serwisowania pojazdu, które zostały przeprowadzone.

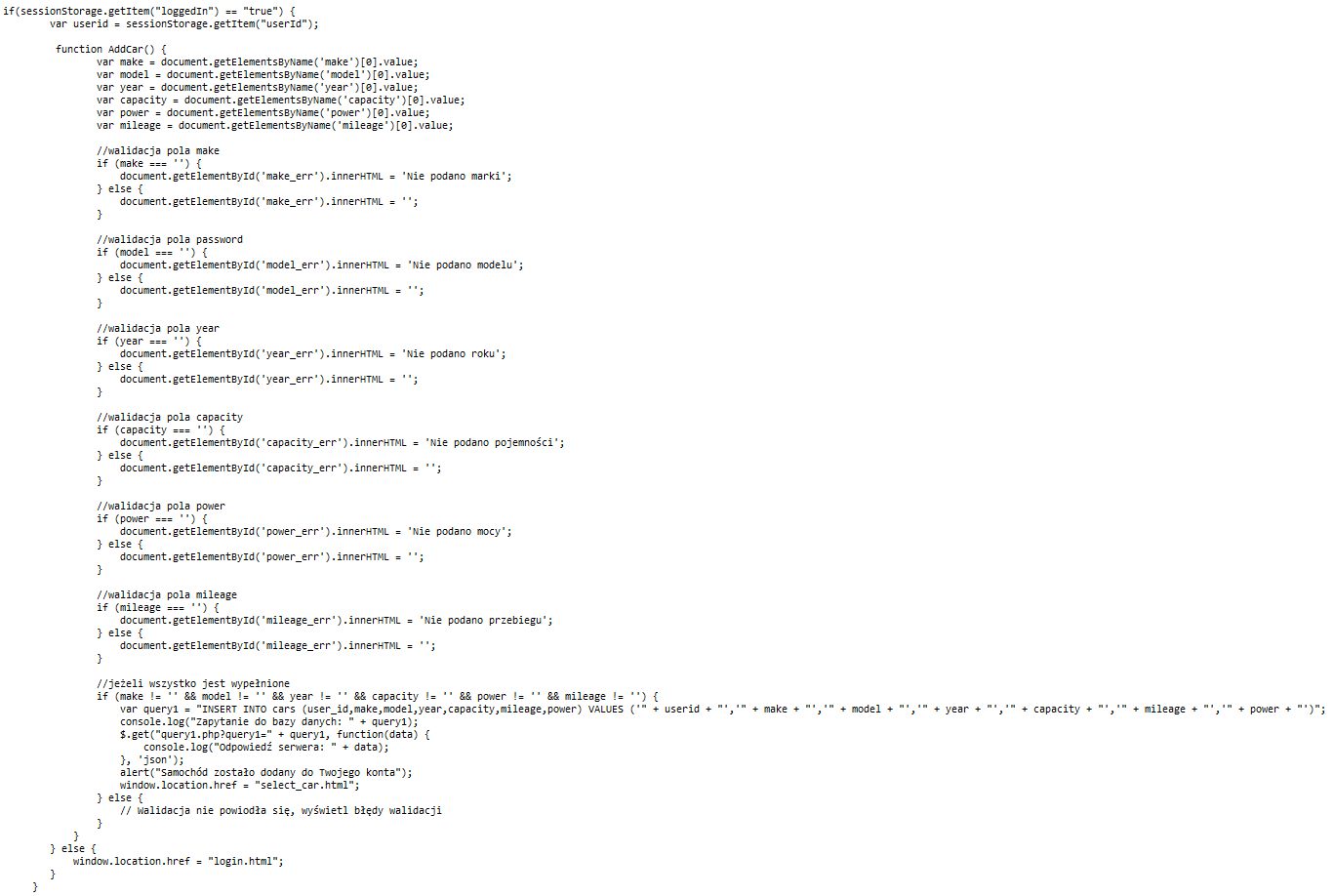
* 1. **Aktualizacja danych**



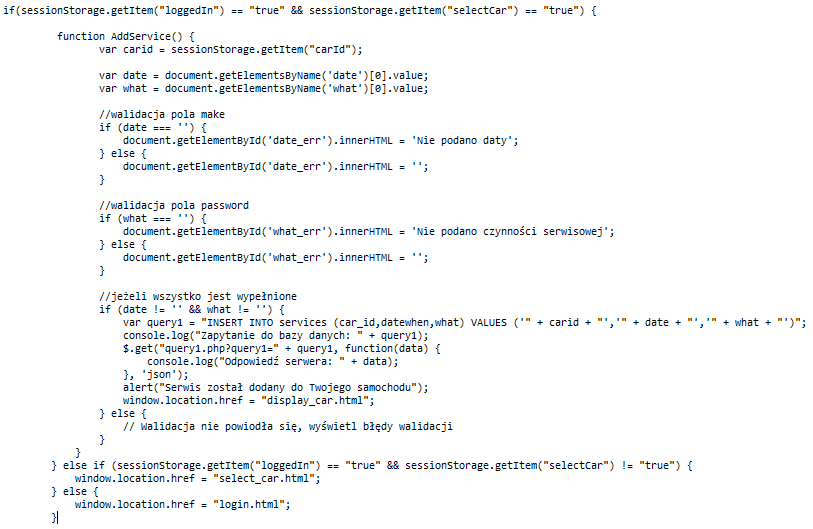
*Screenshot : Kod aktualizacji przebiegu w kodzie JavaScript*

Przedstawiony kod odpowiada za weryfikację wprowadzonych przez użytkownika danych i porównanie ich z wartością elementu (przebiegu) zapisanego w bazie danych. W przypadku, gdy użytkownik poda wartość nie większą niż aktualny przebieg pojazdu, przebieg ten nie zostanie zaktualizowany.

* 1. **Dodanie elementu**



*Screenshot : Kod dodania serwisowania w kodzie JavaScript*



*Screenshot : Kod dodania samochodu w kodzie JavaScript*

Zarówno pierwszy, jak i drugi przedstawiony kod odpowiadają za dodawanie nowych elementów do bazy danych. W pierwszym przypadku dodawane są notatki serwisowe dla konkretnego samochodu i wyświetlane są one w formie listy. W drugim przypadku nowy pojazd zostanie przypisany do profilu aktualnie zalogowanego użytkownika i również zostanie wyświetlony w formie listy.

1. **Testy wydajnościowe**
   1. **Metodologia testowania**
   2. **Narzędzia do testowania wydajności**
   3. **Wyniki testów i ich analiza**
2. **Podsumowanie i wnioski**
   1. **Porównanie wydajności aplikacji w PHP i JavaScript**
   2. **Ocena projektu aplikacji webowej do serwisowania samochodów osobowych**
   3. **Propozycje rozwoju i usprawnień aplikacji**

**Zawartość płyty**

**Bibliografia i netografia**