# Dobro & Dudys – gabinet stomatologiczny

|  |  |
| --- | --- |
| **Dokument:** | Dobro & Dudys – gabinet stomatologiczny |
| Autor: | Michał Dudys, Natalia Dobrowolska |
| Wersja: | 1.0 |
| Data: | 28.05.2019 |
| Status: | Do sprawdzenia |
| Poufność: | Klient/poufne |

Spis treści

[*1. Dobro & Dudys – gabinet stomatologiczny 1*](#_Toc10061391)

[*2. Informacje ogólne 3*](#_Toc10061392)

[*3. Wymagania funkcjonalne 4*](#_Toc10061393)

[*3.1. Wymagane technologie 4*](#_Toc10061394)

[*3.2. Wymagane funkcjonalności strony 4*](#_Toc10061395)

[*4. Responsywność 5*](#_Toc10061396)

[*5. Jakość kodu 5*](#_Toc10061397)

[*6. Walidacja HTML/CSS 7*](#_Toc10061398)

[*6.1. Walidacja HTML 7*](#_Toc10061399)

[*6.1. Walidacja CSS 7*](#_Toc10061400)

[*7. Użyteczność 8*](#_Toc10061401)

[*8. Testowanie 18*](#_Toc10061402)

# Informacje ogólne

Strona gabinetu stomatologicznego Dobro & Dudys powstała na rzecz projektu zaliczeniowego z przedmiotu „Programowanie aplikacji internetowych” pt.: „System rejestracji wizyt dla gabinetu stomatologicznego, gdzie pracuje kilku stomatologów”. Przedmiot prowadzony jest przez Tomasza Nowackiego w Wyższej Szkole Bankowej w Poznaniu.

Projekt podjęty został przez zespół złożony z dwóch osób grupy K27 studiów niestacjonarnych kierunku „Informatyka” – Natalii Dobrowolskiej – aspirującego grafika wektorowego/front-end developera oraz Michała Dudys – software developera lubującego się w zagmatwanym back-endzie oraz bardziej uporządkowanym front-endzie.

Strona powstała na przełomie kwietnia oraz maja 2019 roku, podczas pozostałego okresu półtorej miesiąca do ostatecznego terminu – doszlifowywana była o dodatkowe funkcjonalności, które pozwoliły rozwinąć zespołowi swoją wiedzę na temat Javascriptu, PHP, HTML, CSS oraz MySQL (nie wliczając w to zmagań z ustawieniami darmowych hostingów).

# Wymagania funkcjonalne

## Wymagane technologie

Założenia projektu zaliczeniowego „System rejestracji wizyt dla gabinetu stomatologicznego, gdzie pracuje kilku stomatologów” opiera się na wykorzystaniu poniższych technologii:

* HTML
* CSS
* PHP
* Baza danych MySQL, MariaDB lub SQL Server
* Opcjonalny JavaScript

Podczas tworzenia naszej strony wykorzystaliśmy wszystkie z wyżej wymienionych technologii, a jako docelową bazę danych wybraliśmy MySQL z racji ogólnodostępnych, darmowych hostingów wykorzystujących *phpAdmina*, a co za tym idzie - MySQL.

## Wymagane funkcjonalności strony

Do pełnego, właściwego funkcjonowania strona wymagała:

* Stworzenia formularza obsługującego zapisywanie się pacjentów na wizyty
* Zabezpieczenie formularza
* Estetycznego, przejrzystego front-endu
* Więcej niż jednego lekarza do wyboru
* Zapisu danych wprowadzonych przez użytkownika do bazy danych
* Opcjonalnego wykorzystania danych na stronie

Wszystkie powyższe wymogi zostały spełnione oraz przetestowane przez grupę pięciu testerów, na podstawie czego strona zyskała wiele poprawek oraz była udoskonalana.

# Responsywność

Strona od samego początku z założenia tworzona była tylko i wyłącznie z myślą o rozdzielczościach monitorów, laptopów oraz tabletów.

Responsywność sprawdzona została na przeglądarkach:

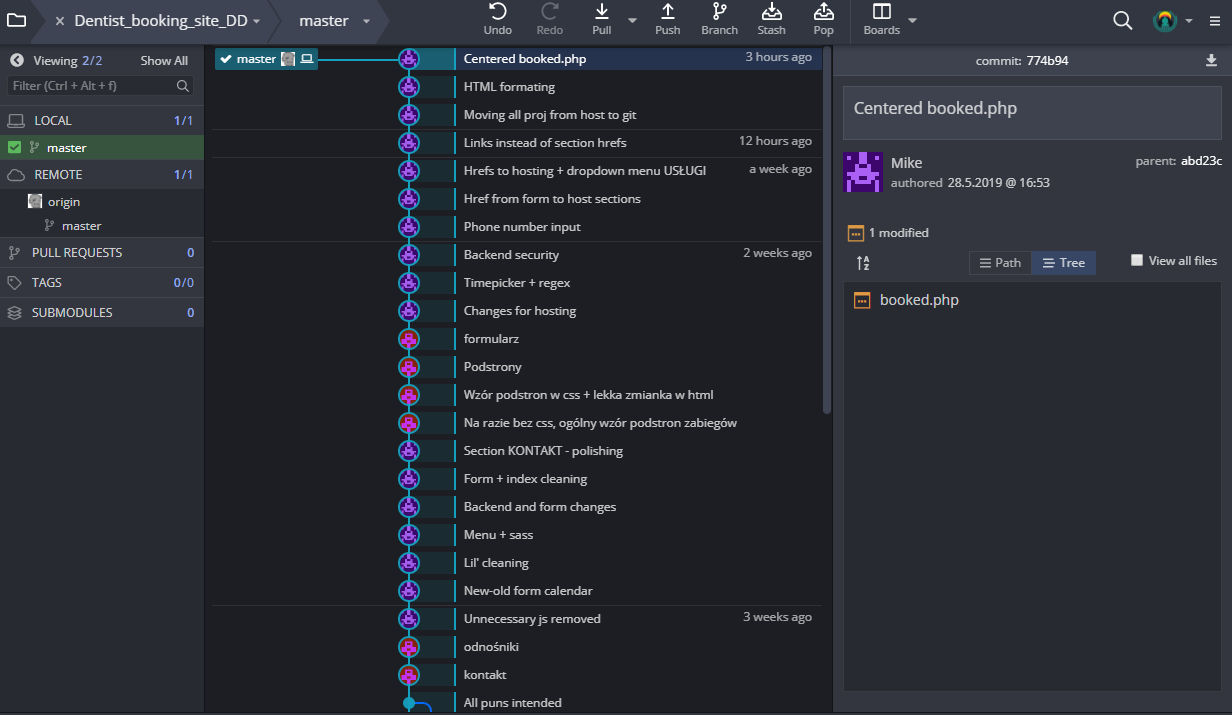
* Chrome
* Opera
* Mozilla Firefox
* Internet Explorer

Na pierwszych trzech przeglądarkach projekt działał odpowiednio, dopiero przy użyciu Internet Explorer pojawiły się pewne niedociągnięcia związane z formularzem, m.in. wpisywanie daty.

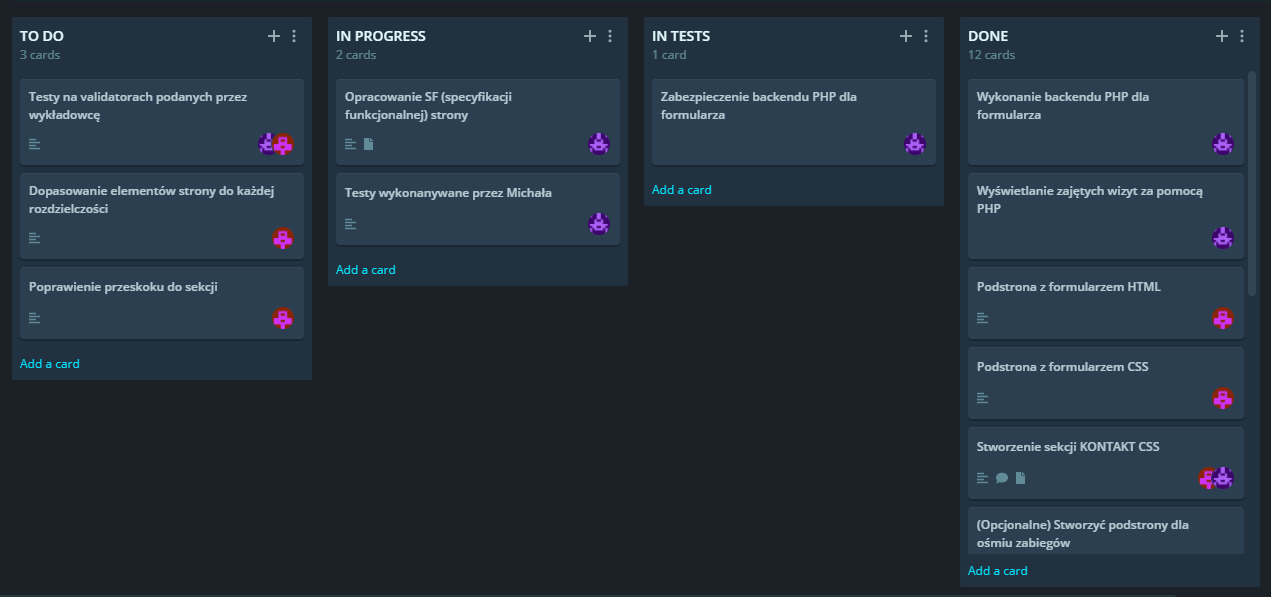
# Jakość kodu

Jakość kodu wielokrotnie sprawdzana była przez testerów oraz członków zespołu. Umożliwił nam to GitKraken – program graficzny podpięty pod repozytorium GitHuba.

Dzięki GitKrakenowi mogliśmy kontrolować nasze wspólne commity, jak i jakość samego kodu, a przy okazji testować najnowsze zmiany z pull requesta drugiego członka zespołu:



Nasze prace zostały również szczegółowo i skrupulatnie rozpisane za pomocą funkcjonalności GitKrakena – Glo - na pomniejsze tickety, dzięki którym całość projektu zrobiona została metodycznie, po kolei, a co najważniejsze – poprawnie:



# Walidacja HTML/CSS

## Walidacja HTML

Walidacja kodu od strony HTML wykonana została za pomocą zaproponowanej przez osobę prowadzącą przedmiot strony – <https://validator.w3.org>.

Dzięki w/w walidatorowi kodu wyeliminować mogliśmy błędy, które umknęły nam podczas testowania i sprawdzania swoich commitów.

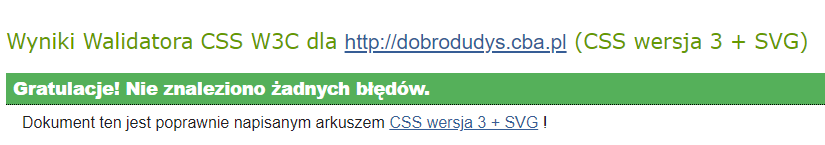
Po ostatecznym „przepuszczeniu” kodu przez walidator pozostały jedynie błędy wygenerowane przez stopkę i nagłówek narzucone przez darmowy hosting CBA.pl oraz niegroźne ostrzeżenia pokroju powtarzającego się ID w elemencie „input”. Zostało to zrobione w ten sposób, by skrócić pliki JavaScriptu z 50 do 10/15 linii kodu, a w dodatku to ID nie jest i nie będzie wykorzystywane w niczym innym, prócz tego skryptu.

Reszta ostrzeżeń walidatora tyczy się wykorzystania spacji w href’ach, co nie wpływa w żaden sposób na działanie strony. Walidator zachęca nas również do stosowania nagłówków h2-h6 w każdej sekcji, nawet jeśli nie jest to potrzebne.

## Walidacja CSS

Do walidacji naszego kodu CSS wykorzystaliśmy walidator https://walidator jihsaw.w3.org

Po sprawdzeniu – nasz kod okazał się być napisany poprawnie:

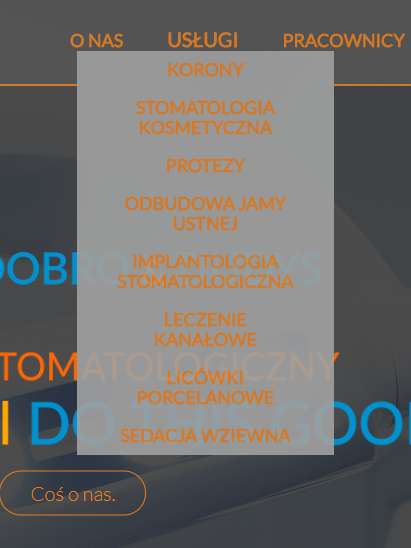


# Użyteczność

Strona spełnia wszystkie założone wymagania. Za pomocą prostego menu możemy dostać się do każdej sekcji wybranej przez użytkownika:

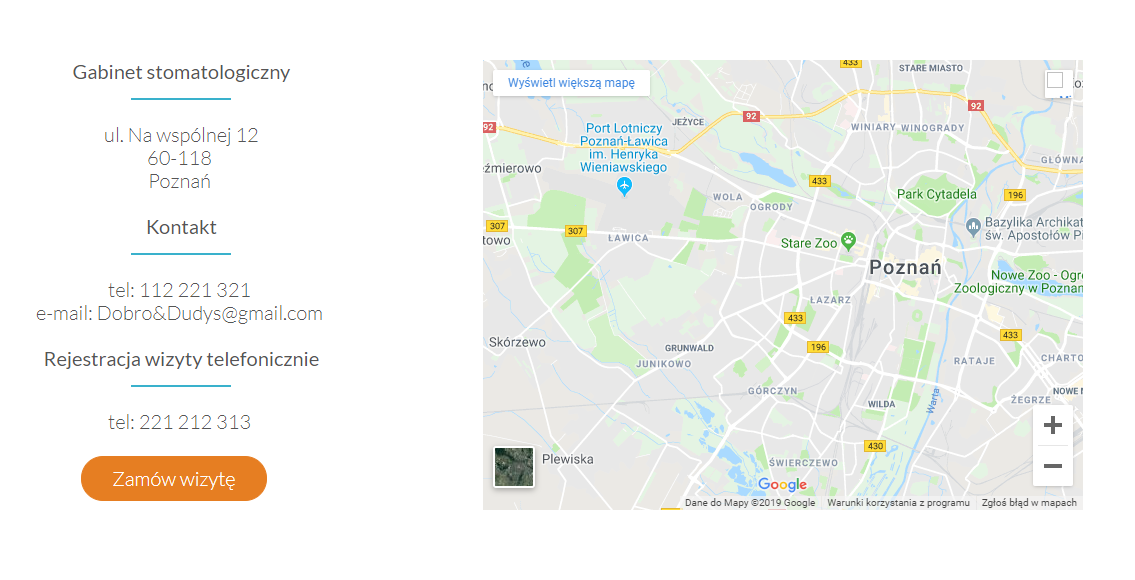


Gdy użytkownik zechce poznać szczegóły usług prowadzonych przez nasz gabinet stomatologiczny, wystarczy, że najedzie na opcję *USŁUGI,* dzięki czemu ukaże mu się rozwijalne menu:

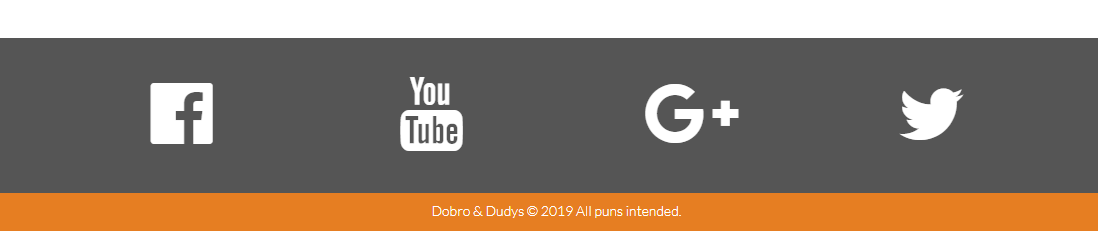


Każda z usług prowadzi do osobnej podstrony z opisem.

Po wybraniu opcji *KONTAKT* użytkownik przeniesiony zostanie do sekcji z danymi naszego gabinetu stomatologicznego:



W stopce znajdują się cztery odnośniki prowadzące (w teorii) do portali społecznościowych naszego gabinetu:



W celu zamówienia wizyty zrobić to można z poziomu trzech miejsc na stronie.

1. Za pomocą opcji na menu, które podąża za użytkownikiem przez całą stronę:



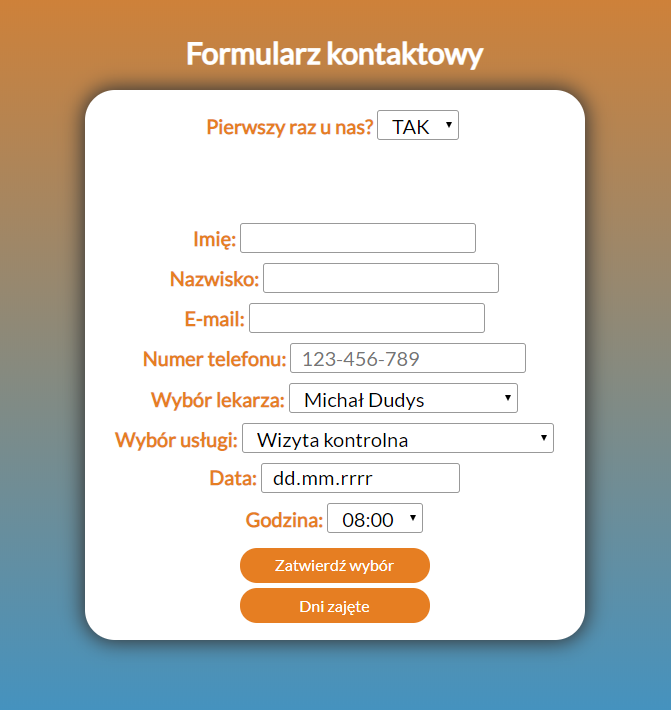
1. Za pomocą przycisku *Umówmy się* w sekcji powitalnej *header*:



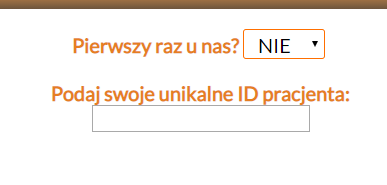
1. Za pomocą przycisku *Zamów wizytę* z sekcji KONTAKT:



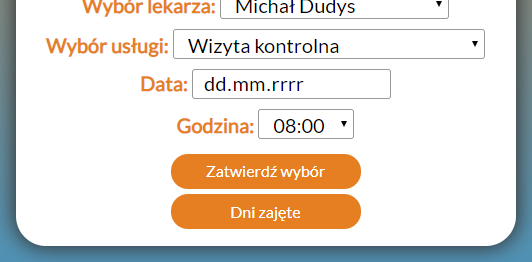
Po wybraniu jednej z trzech opcji użytkownik przeniesiony zostaje do podstrony z formularzem:

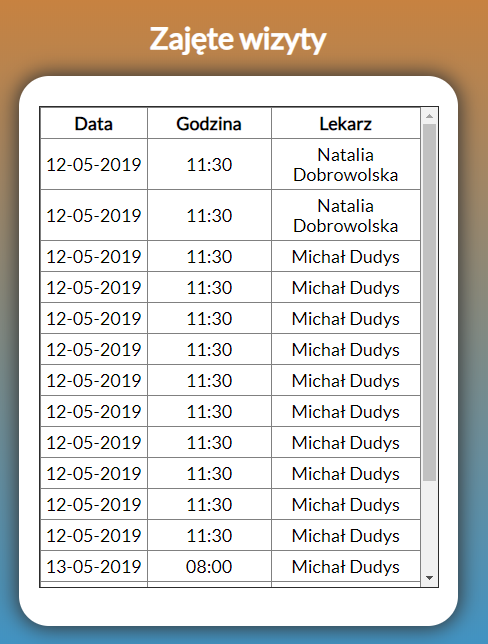


Jeśli jest to pierwsza wizyta użytkownika – musi on podać wszystkie swoje dane oraz wybrać lekarza, datę, usługę oraz godzinę wizyty. Jeśli jednak użytkownik istnieje już w naszej bazie danych – wybrać on musi opcję NIE z listy *Pierwszy raz u nas?* Po wybraniu tej opcji pojawi się pole do wpisania unikalnego ID Klienta (wysyłanego wiadomością email na podany adres przy pierwszym zamawianiu wizyty), dzięki któremu nie trzeba ponownie wpisywać wszystkich swoich danych:

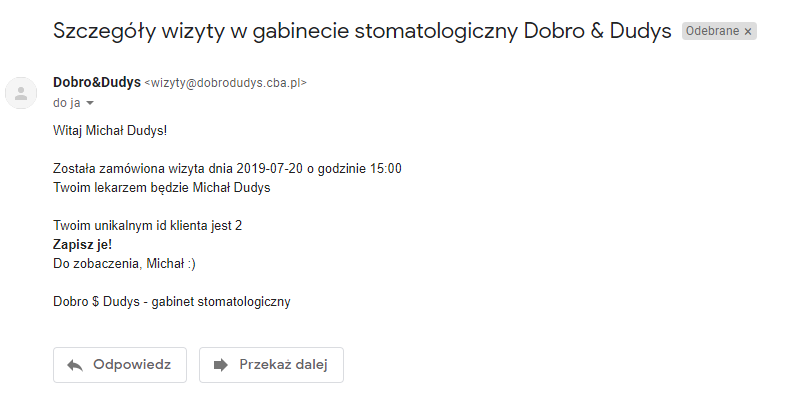


W celu upewnienia się, czy wybrana przez użytkownika data, godzina oraz sam lekarz są dostępne – stworzony został przycisk *Dni zajęte* pokazujący listę umówionych wizyt na dany dzień, godzinę oraz z uczestniczącym lekarzem (tylko aktualne, brak wglądu w wizyty o dacie mniejszej od aktualnej):

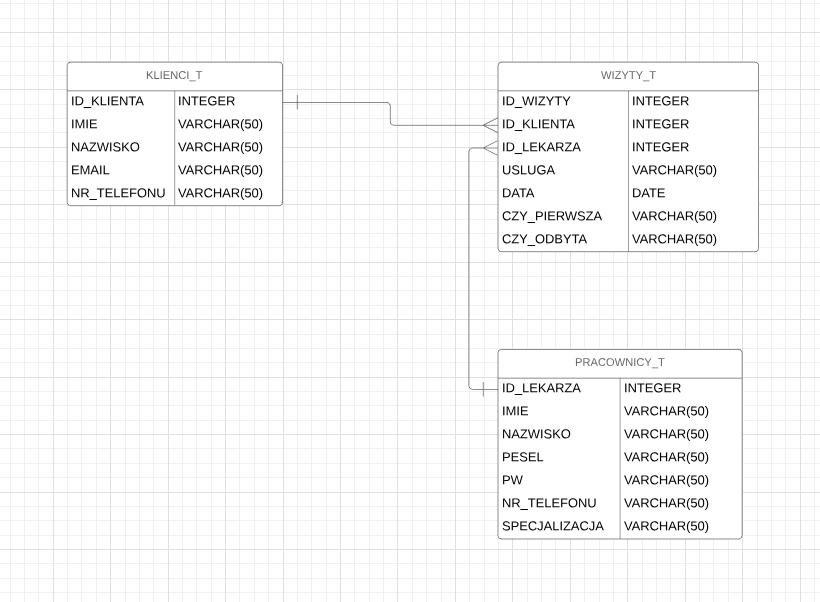




Po udanej rejestracji wizyty do użytkownika zostanie wysłany e-mail na podany w formularzu adres. W wiadomości otrzymamy szczegóły wizyty oraz unikalne ID Klienta. Przykładowa wiadomość:



Wszystkie informacje przekazane poprzez API zostają zapisywane w bazie danych MySQL. Baza zrobiona została na podstawie poniższego schematu:



# Testowanie

W celu poprawnego przetestowania strony należy użyć adresu <http://dobrodudys.cba.pl>, który zaprowadzi do projektu. Zalecam sprawdzenie wszystkich funkcjonalności z rozdziału UŻYTECZNOŚĆ.

W dodatku polecam sprawdzenie zabezpieczeń backendu formularza. Zabezpieczenia opierają się na sprawdzeniu, czy podane ID Klienta jest poprawne, czy dany numer telefonu nie występuje już w bazie, czy dany adres e-mail nie występuje już w bazie.

Zabezpieczenia zapobiegają również zamówienia wizyty u danego lekarza, który wybranego przez użytkownika dnia oraz godzinie posiada już inną, umówioną wizytę.

W tym celu zostało przygotowane pięć umówionych wizyt:



Oraz stworzone 5 użytkowników o danych:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID klienta** | **Imię** | **Nazwisko** | **Email** | **Numer telefonu** |
| 18 | a | a | a@a.a | 123-123-123 |
| 19 | b | b | b@b.b | 111-111-111 |
| 20 | c | c | c@c.c | 222-222-222 |
| 21 | d | d | d@d.d | 333-333-333 |
| 22 | e | e | e@e.e | 444-444-444 |

Po udanej próbie dodania wizyty wyświetli się komunikat:



Jeśli jednak zabezpieczenie zadziała, to otrzymamy komunikat z tekstem błędu (zmieniającym się w zależności od potrzeby):



Cały projekt został stworzony na darmowym hostingu [www.cba.pl](http://www.cba.pl). Poniżej udostępniam hasła w przypadku chęci zajrzenia do plików oraz bazy danych:

* Logowanie do strony:
  + L: Dojdyl
  + H: 1Q0@2m9jot5w4h7y<34
* Baza danych:
  + L: Dojdyl
  + H: 1Q0@2m9jot
* FTP:
  + Dane takie same, jak w przypadku logowania do strony
* E-mail:
  + L: [wizyty@dobrodudys.cba.pl](mailto:wizyty@dobrodudys.cba.pl)
  + H: 1Q0@2m9jot5w4h7y<34

|  |  |  |  |
| --- | --- | --- | --- |
| **Zmiany** | **Wersja** | **Data** | **Osoba** |
| Utworzenie | 1.0 | 28.05.2019 | Michał Dudys |
| Aktualizacja informacji o funkcjonalności wysłania emaili do userów | 1.1 | 29.05.2019 | Michał Dudys |
|  |  |  |  |