

Uwaga!

Pamiętaj, że pliki PHP należy uruchamiać za pośrednictwem serwera Apache.

Skopiuj zatem cały folder – np. **ćwiczenie1** do katalogu **C:\xampp\htdocs** a następnie podgląd działania pliku uruchomisz w przeglądarce pod adresem: **http://localhost/ćwiczenie1** lub **http://127.0.0.1/ćwiczenie1**

Zwróć też koniecznie uwagę, aby **zmian dokonywać we właściwych plikach** – upewnij się, że w edytorze kodu otwarty został skrypt PHP znajdujący się w podfolderze **htdocs** serwera, a nie na przykład leżący gdzieś na pulpicie...

Analogicznie postępuj w ćwiczeniu drugim – katalog **ćwiczenie2**

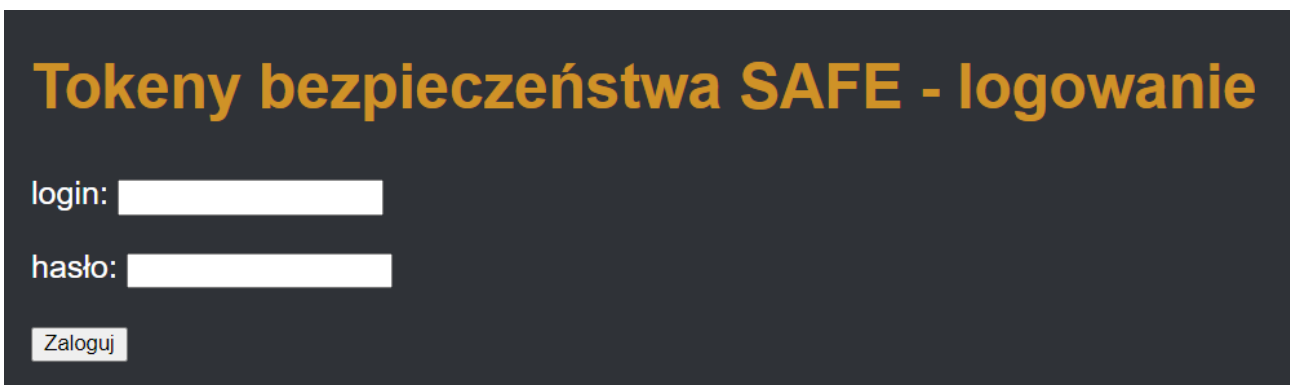
Ćwiczenie 1 → Folder: ćwiczenie1

Paweł i Gaweł w jednym pracowali banku. Paweł (pracujący na parterze) zarządza działem promocji, zaś Gaweł (zamieszkujący piętro) zasadniczo jest szefem wszystkich szefów.

Napisz w PHP **webowy system logowania**, do którego dostęp powinni mieć tylko obaj panowie – wejście do systemu możliwe jest TYLKO na dwa konta:

- Paweł → login: **pawlo** hasło: **napadnabank**
- Gaweł → login: **gawlo** hasło: **likeaboss**

Formularz logowania w pliku **index.php** prezentuje się następująco:



The image shows a login form with a dark background. At the top, the title "Tokeny bezpieczeństwa SAFE - logowanie" is displayed in a large, bold, orange font. Below the title, there are two input fields: "login:" followed by a white text box, and "hasło:" followed by a white text box. At the bottom left, there is a button labeled "Zaloguj" in a small, light gray box.

Jeżeli użytkownik witryny użyje któregoś z podanych powyżej zestawów danych dostępowych, to po zalogowaniu się otrzyma specjalny **10-znakowy token bezpieczeństwa**, który pozwoli mu dostać się do skarbca w faktycznej siedzibie banku na Manhattanie, w dzielnicy Wall Street.

Tokeny bezpieczeństwa dostarcza firma SAFE – akronim od Super Awesome Fast Engine (zupełnie przypadkowo jest to działalność gospodarcza prowadzona przez kuzyna Pawła). **Własna funkcja w PHP o nazwie token()** już znajduje się w źródle, generując dla użytkowników działające tokeny.

Niemniej jednak należy sprawić, aby taki token dostępu do skarbca bankowego **pojawiał się TYLKO po poprawnym zalogowaniu użytkownika!**

Przykładowe wykonanie skryptu **dla poprawnych danych logowania Pawła** wprowadzonych do formularza:

Witaj Paweł, weź się do roboty leniu xD

Jednorazowy token dostępu do skarbca: UsCqBaiZRn

Przykładowe wykonanie skryptu **dla poprawnych danych logowania Gawła** wprowadzonych do formularza:

Witaj Gaweł, szefie wszystkich szefów!

Jednorazowy token dostępu do skarbca: 3X9UAaLI9v

Natomiast dla wszystkich innych danych logowania, rezultatem wysłania formularza powinien być komunikat:

Niepoprawny login lub hasło!

Ten sam skrypt **index.php** ma zająć się danymi przesłanymi metodą POST z formularza, witryna nie może też wyświetlać żadnych widocznych ostrzeżeń przy wejściu na stronę, gdyż inaczej szefa „szlag trafi” ze złości na felerny system na który wydał miliony dolarów.

Dane dostępowe nie mają być pobierane z bazy danych, tylko wprost wpisane w instrukcje warunkowe w skrypcie.

Ćwiczenie 2 → Folder: cwiczenie2

Malwina postanowiła wykorzystać swoje znaczne umiejętności dydaktyczne do zarobienia pieniędzy na nowy samochód – zamierza oferować w internecie zdalne korepetycje dla przyszłych maturzystów i studentów.

Malwina zwróciła się do Ciebie o pomoc w zrobieniu strony www, na której ludzie mogą umawiać się na zajęcia zdalne. Sama nie potrafiła zrealizować w kodzie nic poza podstawowym szablonem HTML (sama jest raczej specjalistką od przedmiotów humanistycznych) – rezultaty jej pracy znajdziesz w folderze o nazwie **cwiczenie2**

Nie ukrywajmy, że Malwina liczy na Twoją pomoc – założenia interfejsu formularza są następujące:

- Użytkownik podaje swoje imię (pole wymagane);
- Z listy wyboru można wybrać jeden przedmiot szkolny z następującego zestawu: język polski, historia, wiedza o społeczeństwie. Jedna godzina lekcji kosztuje odpowiednio:
 - 50 zł język polski,
 - 60 zł historia,
 - 45 zł wiedza o społeczeństwie;
- W polu numerycznym użytkownik wpisuje liczbę zamówionych lekcji – musi to być wartość niepusta – musi to być liczba całkowita dodatnia;
- Dodatkowo, można u Malwiny wykupić notatki z zamówionych lekcji – jest to dodatkowy koszt +5 zł za każdą godzinę zajęć i jest to decyzja jednorazowa, czyli nie można wziąć notatek do np. połowy lekcji – albo się decydujesz na wszystkich zamówionych godzinach, albo w ogóle
- Podsumowanie formularza ma wyświetlić całkowity koszt zamówionych zajęć, o ile dane podane w formularzu pozwalają taki koszt obliczyć (są poprawne)
- Daty i godziny zajęć póki co nie określamy – Malwina chce się najpierw zapoznać z wersją beta, umożliwiającą wybór przedmiotu, liczby godzin oraz opcjonalnym wyborem otrzymania notatek w cenie 5zł za każdą lekcję. Jak również oczywiście z wersją obliczającą już teraz łączny koszt zamówienia.

Skrypt **index.php** ma zawierać tylko interfejs formularza, zaś podsumowanie dokonanego zamówienia korepetycji ma zostać wyświetlone w zewnętrznym pliku **korepetycje.php**

Jeżeli nie wypełniono i nie przesłano formularza to skrypt **korepetycje.php** powinien automatycznie przekierować użytkownika do pliku z formularzem, czyli do **index.php**

Skrypt **korepetycje.php** powinien walidować (sprawdzić) poprawność danych przysłanych żądaniem POST przez klienta i nie wyświetlać obliczonej łącznej kwoty, jeżeli jej obliczenie nie jest możliwe.

Screenshotów interfejsu formularza ani przykładowego jego wykonania nie podano, aby całkowicie samodzielnie wykonać od zera projekt – jedne czym dysponujemy, to informacje dostarczone przez zleceniodawcę – Malwinę.