1.

**А)**

GET — получение ресурса

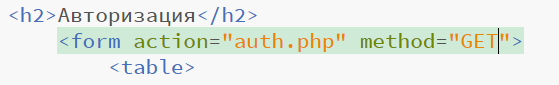
POST — создание ресурса

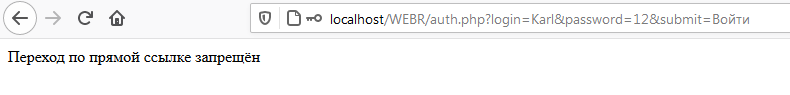
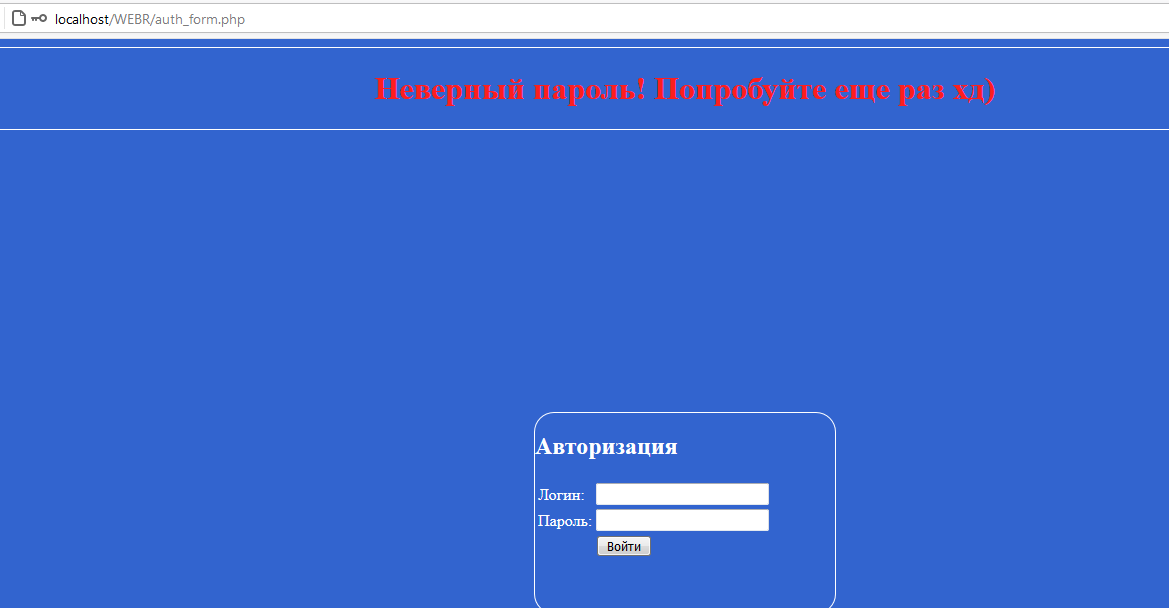
PUT — обновление ресурса

DELETE — удаление ресурса

Очевидно, что из всего этого списка нам нужен только POST-запрос, который будет создавать ассоциативный массив с паролем, логином и кнопкой. После уже отправлять в нужный файл и обрабатывать.  
**B)**

Если попробуем GET-запрос, то получим неожиданный текст после попытки авторизации под неправильным паролем



  
Вот что он должен был вывести (POST-запрос):  


2.

**А)**

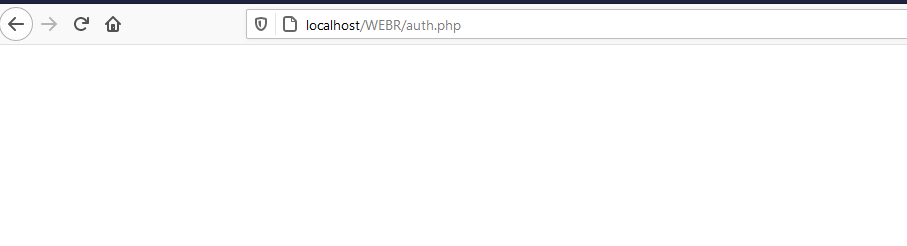
Хеш-функция это **функция**, осуществляющая преобразование массива входных данных произвольной длины в (выходную) битовую строку установленной длины, выполняемое определённым алгоритмом. Это используется для шифрования паролей (более безопасного их хранения в базе данных, в случае взлома базы).  
Одна из таких функций – md5(<пароль>).

На выходе мы получим вместо ‘1234’ что-то типо: C93D3BF7A7C4AFE94B64E30C2CE39F4F

**В)** Данные функции не применяли, но к следующей лабе постараюсь их применить.

3.

По идее, мы можем обратиться к файлу auth.php напрямую, не нажимая кнопку “ВОЙТИ”.

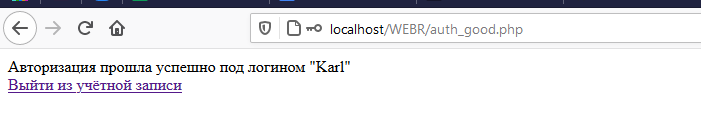
  
Для того, чтобы исправить это, нам нужна проверка на нажатие кнопки “ВОЙТИ”.

Выглядеть это будет вот таким образом:  
  
Если была нажата кнопка -> проверяем логин и пароль  
Иначе (если кнопка не нажата) -> выводим сообщение о том, что мы пытались перейти на сайт auth.php незаконным способом ☺

4.

Если я правильно понял, то нам нужно избежать возможности авторизации, если мы уже авторизованы. Данный феномен я сразу предусмотрел и сделал проверку на «авторизован ли пользователь в данный момент» внутри файла с самой формой:



Если мы авторизованы, он отправит нас вот на эту страницу:  


Не особо креативно, но всё-таки ☺  
Здесь уже можно будет выйти из учётной записи и авторизоваться под другим аккаунтом.

5.  
1. Авторизация на реальных сайтах прежде всего использует капчу, которая позволяет избежать самого простого бота, который будет тупо перебирать пароли к логину.

2. Так же там использую хеш-функцию md5() для шифрования паролей + «соль», т.е. добавление к шифрованию пару символов и опять шифруем…  
  
3. Проверка на то, сколько раз мы пытались зайти на сайт. Если к примеру мы попытались зайти раз 10, то уведомляется пользователь под тем логином, которого пытаются взломать.