# **Тестовое задание**

**Использовать:**

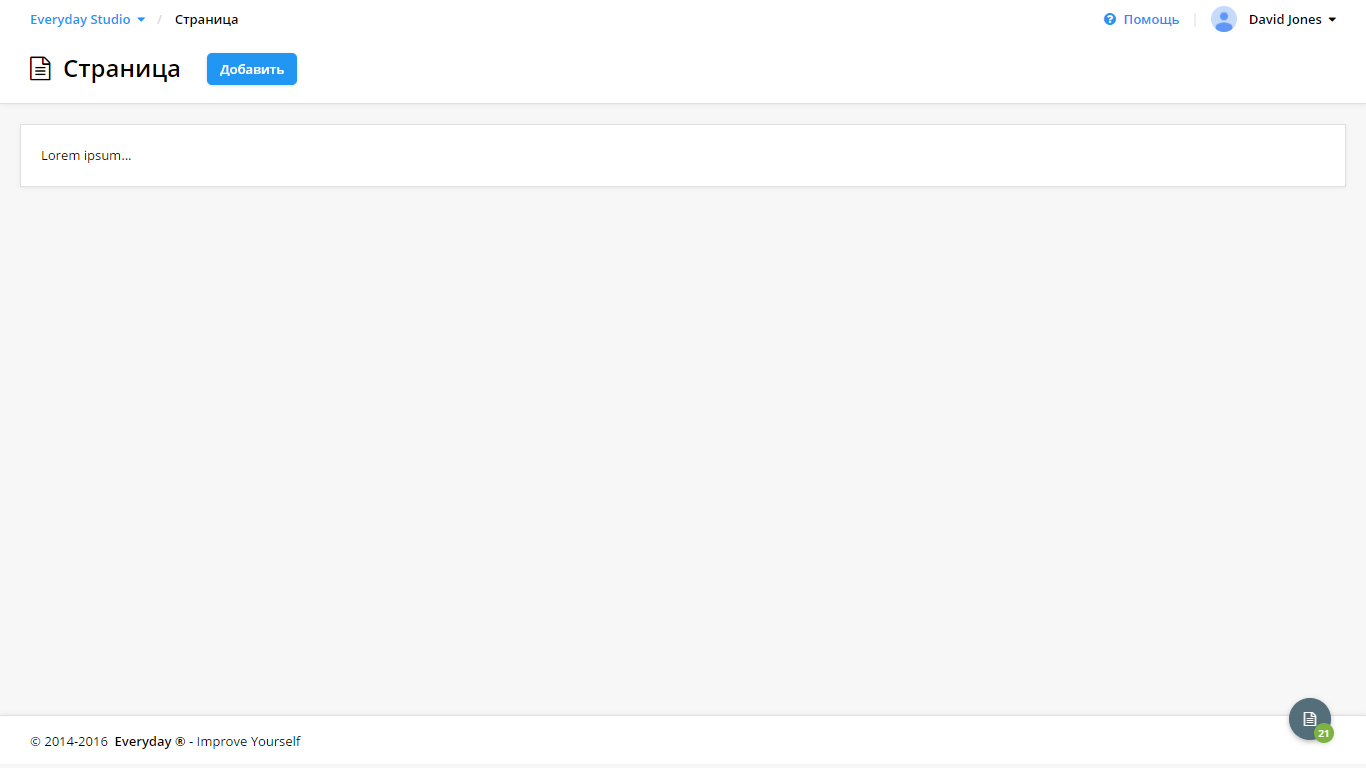
* [Normalize.css](https://necolas.github.io/normalize.css/)
* [JQuery](https://ajax.googleapis.com/ajax/libs/jquery/2.2.2/jquery.min.js)
* [Font Awesome](http://fortawesome.github.io/Font-Awesome/)
* [Open Sans](https://www.google.com/fonts#UsePlace:use/Collection:Open+Sans)
* [jQuery Cookie](https://github.com/carhartl/jquery-cookie)

**Стиль написания кода:**

* отступ в 1-ну табуляцию
* одинарные кавычки в PHP
* двойные кавычки в JS|CSS|HTML
* кодировка UTF-8 без BOM

## **Сверстать структуру страницы**

Взять прикрепленный к заданию PSD-шаблон (изображение) и сверстать страницу придерживаясь внешнего вида 1:1. При этом весь проект должен иметь разбитую иерархию и структуру файлов. Все ссылки и кнопки должны менять цвет при наведении, а поля (раздел **1.4**) менять цвет ободка при фокусе.

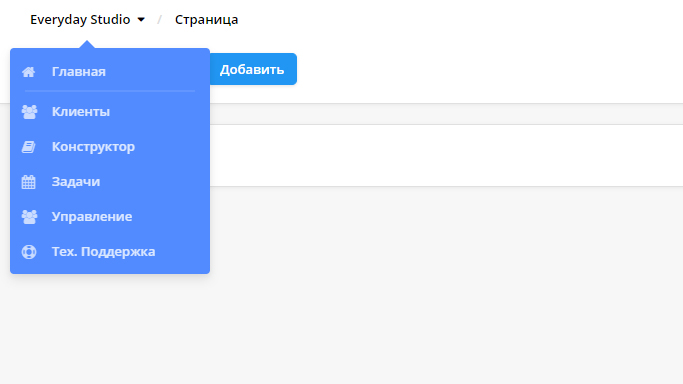


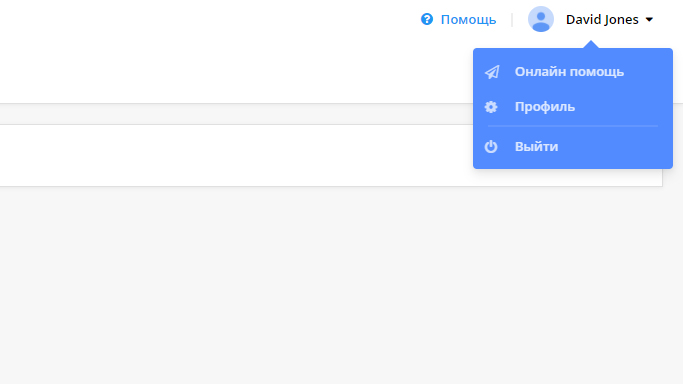
## **Создать сессию**

При заходе на страницу генерировать сессию средствами php. В сессию поместить имя (David Jones) закодированного через **md5** и указать время жизни в 15 минут – это будет токен. Возле имени (David Jones) вывести время жизни которое осталось у сессии.

## **1.3 Сверстать 2 выпадающих списка**

Сверстать 2-а списка слева и справа, которые появляются при наведении на ссылку меню.





## **1.4 Написать php-функцию с масивом чисел**

Написать функцию которая будет генерировать массив случайных чисел в 10 значений вида:

**$numbers = [**

**‘number’ => ‘number’,**

**‘number’ => ‘number’,**

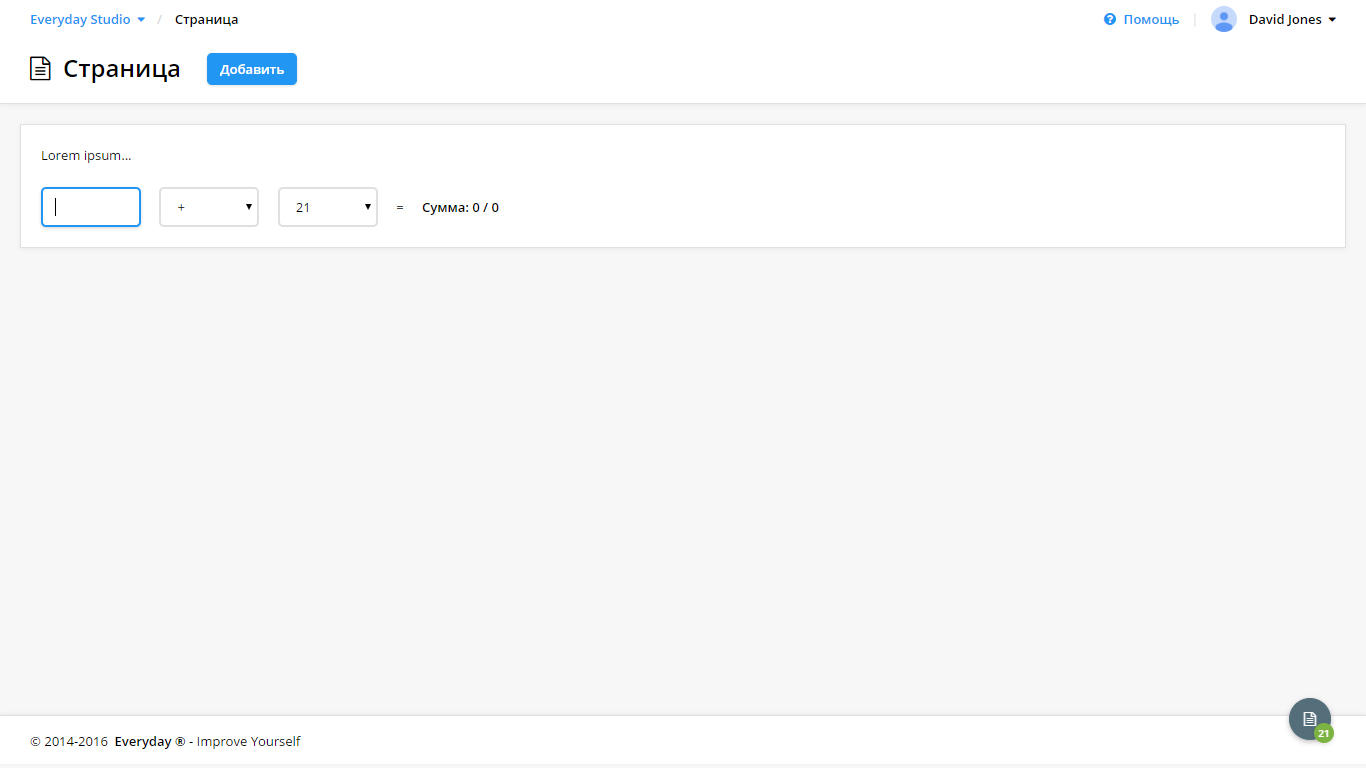
**etc…**

**];**

## **1.5 Создать форму калькуляции**

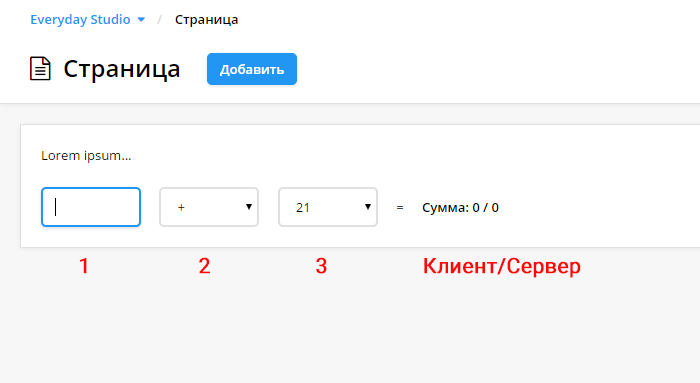
* Создать 1-но текстовое поле формы для внесения числового значения
* Создать 1-ин выпадающий список в котором должно быть 10 пунктов со значениями. Для этого используем созданную функцию из раздела **1.3** и с нее берем значения.
* Создать 1-ин выпадающий список со знаками [ + ] [ - ] [ \* ] [ / ] используя только jQuery и поместить его между текстовым полем и выпадающим списком с числами.
* Создать блок с выводом калькуляции формулы:

(число вписанное в поле №1) (знак выбранный в списке №2) (число выбранное в списке №3) = СУММА



В сумме есть два значения разделенные линией. Первое – сумма посчитанная на клиентской части (с использованием jQuery), второе – сумма посчитанная на серверной части (под серверной частью имеется в виду php-функция из раздела **1.5**).

Сумма должна пересчитывается в реальном времени при изменении любого из полей. Значение на сервере должно выводится через Ajax-запрос. Предусмотреть условие при котором первое поле может быть не заполнено или иметь не числовое значение.



В итоге: при изменении первого/второго/третьего поля мы получаем два значения суммы, одно считается на сервере и подставляется в левую часть суммы, второе считается на сервере с помощью функции и через jQuery-Ajax-Запрос подставляется в правую часть суммы.

## **1.6 Написать php-функцию калькуляции**

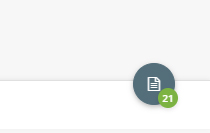
Берем значения с 3 полей и обрабатываем по принципу представленному в разделе **1.4**. Функция должна возвращать сумму и дату (в формате: дд.мм.гггг чч:мм) когда функция вернула ответ (дату определять внутри функции). Использовать только POST-запрос.

## **1.7 Сохранять значение 1-го поля в Cookie**

Сделать возможным сохранять значение текстового (1-го) поля в Cookie и возвращать его в это же поле после перезагрузки страницы.

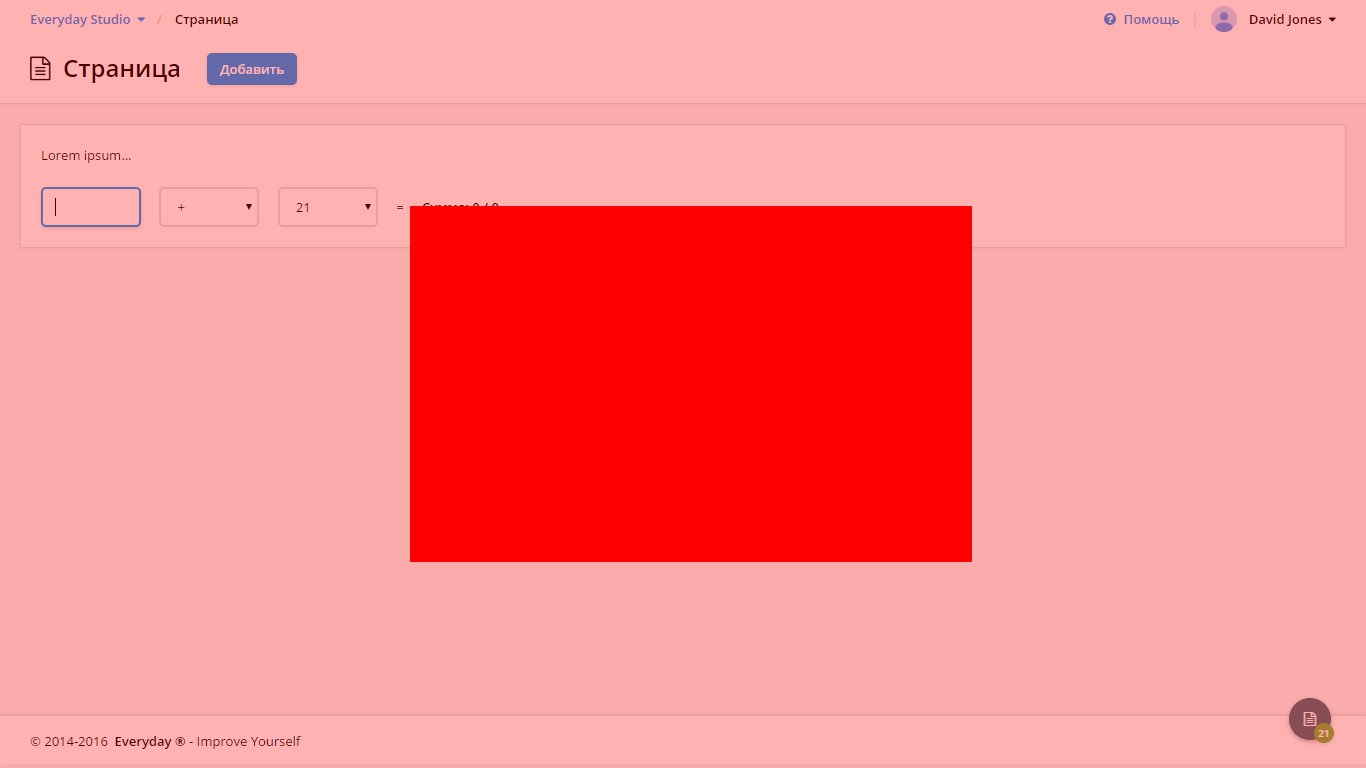
## **1.8 Вести историю калькуляции**

Сделать скрытый блок и туда записывать (в виде списка или таблицы) значение суммы при каждой калькуляции используя jQuery. Количество записей в истории выводить в индикатор (зеленый круг). К каждой сумме должна так же быть привязана дата записи в историю (в формате: дд.мм.гггг чч:мм). В итоге должно получится в записи: Клиентская сумма и дата добавления, серверная сумма и дата добавления.



## **1.9 При нажатии на кнопку выводить всплывающее окно с историей калькуляции**

При нажатии на эту круглую кнопку нужно отобразить скрытый блок по центру страницы, сделать затемнение фона и вывести список произведенных калькуляций. При этом оформить блок в стилистики самой страницы и сделать для него плавное появление (все анимации делать только на CSS). У блока должен быть заголовок с названием (для понимания что перед нами появилось).



При нажатии на фон блок должен так же плавно исчезать.

## **1.10 Запись логов**

При открытии и закрытии окна с помощью php-функции и аякс-запроса (все написать самому) вести логирование вида: Дата (в формате: дд.мм.гггг чч:мм), токен (раздел **1.2**) и статус (открыто окно/закрыто окно) и сохранять его в текстовый файл. Предусмотреть что файл может быть удален и его нужно создать самому перед записью.

## **Примечание**

Во внимание будет браться не только подход к реализации, методики использованные при написании, а и чистота кода, его оптимальность. К документу нужно приложить текстовый документ где указать, операционную систему и редактор кода на котором велась разработка.