

# Тестовое задание на вакансию стажёр веб-разработчик

## Конвертер римских и арабских чисел

Римляне, как известно, использовали для записи числа латинские буквы. Считается, что римская система счисления является классическим примером непозиционной системы счисления, то есть такой системы счисления, в которой величина (которую обозначает цифра) не зависит от положения в числе.

Напомним, что в римской системе счисления I обозначает 1, V обозначает 5, X — 10, L — 50, C — 100, D — 500, M — 1000. Например, число 3 в римской системе счисления будет обозначаться как III.

Однако, на самом деле, не все так просто, и она не является полностью непозиционной системой счисления, потому что в римской системе счисления есть дополнительное правило, которое влияет на величину, которую обозначает цифра, в зависимости от ее положения. Правило это запрещает употребление одной и той же цифры более 3 раз подряд, поэтому три - это III, а четыре - это уже IV, и I(1), стоящая перед большей цифрой V(5), обозначает вычитание, то есть фактически равна -1.

## Задача

Спроектировать и разработать одностраничное веб-приложение, позволяющее конвертировать числа из римских в арабские и наоборот.

## Основные характеристики веб-приложения:

- состоит из клиентской и серверной частей, конвертирование должно проходить на стороне сервера;
- для реализации этой части допускается использование любого языка программирования (Python, Ruby, Java, C#, Php);
- обмен данными между клиентской и серверной частями происходит с использованием формата JSON;
- для хранения данных на сервере используется какое-либо хранилище: реляционная/нереляционная БД, простой файл, оперативная память или др.
- клиентская часть, по возможности, использует любой JavaScript-фреймворк на ваш вкус (AngularJS или ReactJS будет плюсом);

## Основной сценарий использования

Макет представлен ниже. Пользователь вводит римское число в поле 2. Нажимает кнопку Конвертировать, в области 4. В окошке 3 появляется результат (арабское число).

## Дополнительные сценарии

- Пользователь может вводить арабское число в поле 2 и сконвертировать в римское.
- После ввода числа в поле, программа определяет к какой системе относится число (римской или арабской) и, соответственно, конвертирует в нужную систему.

## Клиентская часть

Пользователь взаимодействует с приложением посредством единственной страницы, через которую подаются входные параметры и отображаются выходные данные.

Макет страницы может иметь вид, как на рисунке ниже. Цифрами на изображении указаны следующие элементы управления:

1. Зона для заголовка и короткого описания.
2. Текстовое поле для ввода числа.
3. Текстовое поле для отображения результата (только чтение).
4. Кнопки и возможные дополнительные элементы, позволяющие конвертировать числа.

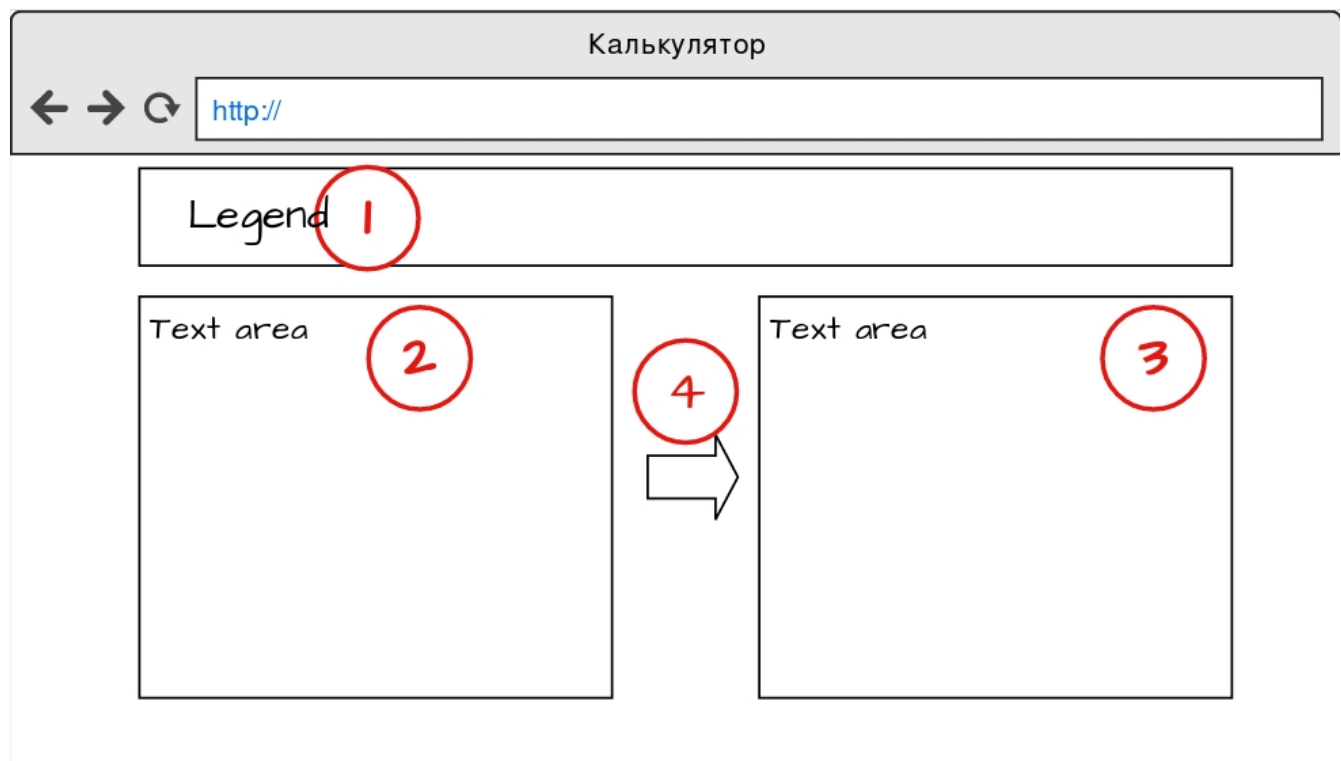


Figure 1: Макет приложения

## Требования к оформлению и критерии оценки

Программа должна быть рабочей. Если присланный код не работает, или его не удаётся запустить из-за отсутствия необходимых инструкций - тестовое задание не рассматривается.

Код должен соответствовать общепринятым стилевым и организационным стандартам, действующим для выбранных вами языков и технологий.

По возможности, код должен сопровождаться разумными комментариями, юнит-тестами, прочими инструкциями.

Обязательным условием является использование систем контроля версии. Проверяться будут только те тестовые задания, исходный код которых находится, например, на github или bitbucket.

Оцениваться тестовое задание будет по следующим критериям:

- За полностью рабочий основной сценарий - максимум 3 балла.
- За каждый полностью рабочий дополнительный - максимум 3 балла.
- За качество кода клиентской части, внешний вид, соответствие общепринятым стандартам - максимум 2 балла.
- За качество кода серверной части и проектирование архитектуры - максимум 2 балла.
- За креативность - 1 балл.

Для прохождения отбора соискатель должен набрать 7 баллов.