

Задачи по веб-разработке

Все мы когда-то начинали свой путь и понимаем, что стажер - это не тот, кто обладает обширными знаниями в выбранной области, а тот, кто хочет и может учиться новому. Именно по такому принципу построены наши задания. Мы описываем реальную проблему, с которой нам приходилось сталкиваться, и предоставляем материалы для ее решения.

Вам остается только усвоить материал (если он вам не знаком), подумать и найти решение. Ничего сложного, никаких специфических знаний от вас не требуется. Только горящие глаза, жажда знаний и немного смекалки.

Желаем удачи!

1. HTML

Руководствуясь приведёнными ниже макетами, создать форму регистрации пользователей (страница “Регистрация пользователя”). При нажатии на кнопку “Отправить” данные должны отправляться на сервер методом POST. Все поля обязательные для заполнения. Нажатие на кнопку “Очистить” сбрасывает все введенные в форму значения. После отправки данных пользователь должен увидеть страницу с подтверждением (страница “Завершение регистрации”).

Регистрация пользователя

Имя:	<input type="text" value="Иван"/>
Фамилия:	<input type="text" value="Петров"/>
Почта:	<input type="text" value="mail@example.com"/>
<hr/>	
Курсы:	<input type="radio"/> Drupal 7 <input checked="" type="radio"/> Drupal 8
Дополнительная информация:	<div>Знание PHP, CSS, Javascript и SQL</div> <div></div>
<div>Отправить</div> <div>Очистить</div>	

Завершение регистрации

Вы успешно зарегистрированы, спасибо!

[Вернуться назад](#) чтобы повторно ввести данные.

На что обратить внимание:

- по условию задачи не требуется какая-либо обработка данных на сервере, т.е. писать php-код не нужно;
- обе страницы должны быть сделаны на [HTML5](#).

2. JS

Разработать [одностраничное приложение](#) (Single page application, SPA), которое будет представлять собой многошаговую форму.

Шаги формы:

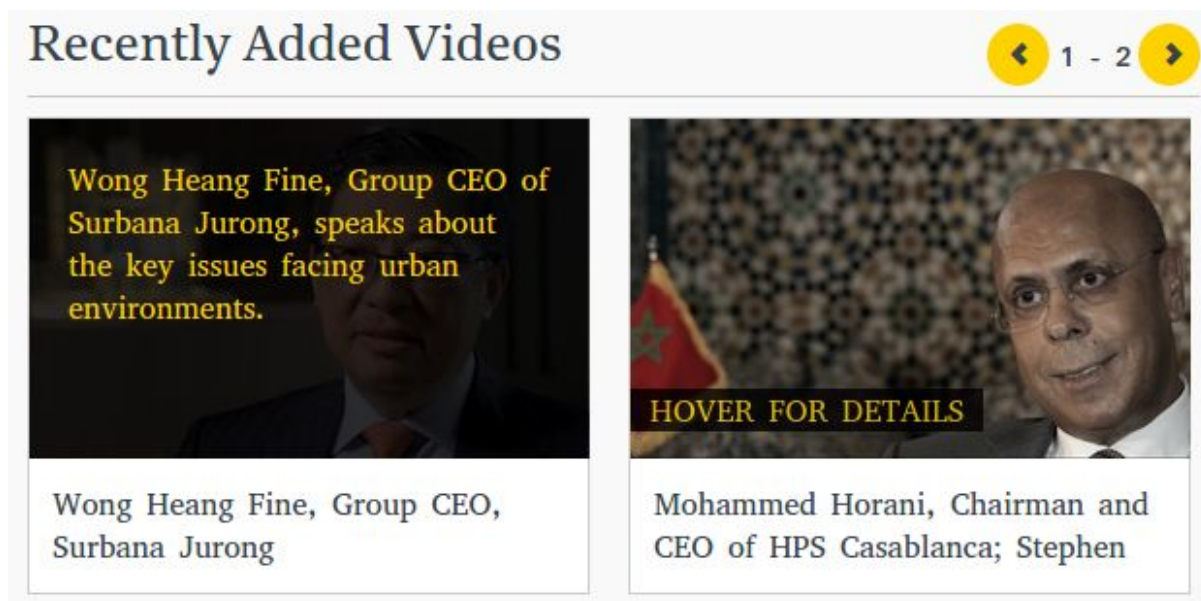
1. пользователь вводит свое имя и жмет кнопку “Далее”
2. пользователь вводит свою фамилию и жмет кнопку “Далее” (он может вернуться на предыдущий шаг, нажав кнопку “Назад”)
3. пользователь вводит свой email и жмет кнопку “Завершить” (он также может вернуться на предыдущий шаг, нажав кнопку “Назад”)
4. появляется сообщение “Спасибо!”

На что обратить внимание:

- все поля формы обязательные для заполнения;
- убедиться, что в поле email находится именно адрес электронной почты, а не случайная строка;
- задание можно сделать и на чистом JS, и с использованием любой библиотеки или любого фреймворка (jQuery, Backbone, Angular, React и т.д.).

3. CSS

Используя [каскадные таблицы стилей](#) (cascading style sheets, CSS), оформить блок с тизерами. Картинки и текст могут быть любыми.



На что обратить внимание:

- при наведении курсора на текст “HOVER FOR DETAILS” на полупрозрачном фоне должен появляться текст с описанием (пример на картинке слева);
- функционал для переключения элементов при нажатии на “<” или “>” не требуется.

4. PHP

Создать страницу и вывести на ней форму поиска и текст, по которому будет производиться поиск. При вводе слова в форму поиска необходимо найти и подсветить все упоминания этого слова в тексте. В случае, если указывается несколько слов, каждое должно искаться индивидуально. Если словосочетание указывается в двойных кавычках, то оно ищется как единое вхождение.

Пример работы такого поиска можно увидеть на приложенном макете. В качестве возможного варианта функции для поиска и замены текста можно использовать [preg_replace](#).

Найти строку в тексте

Ключевая строка:

Поиск

Lorem ipsum dolor **sit** amet, consectetur **adipiscing elit**, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Sed ut perspiciatis unde omnis iste natus error **sit** voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo.

5. SQL

Ниже приведена структура базы данных, в которой отражена зависимость между списком стажёров (таблица “Trainee”) и курсами (таблица “Course”), которые они проходили. Из таблицы “History” можно получить информацию о том, какой стажёр проходил выбранный курс, сколько это заняло времени (поля “from” и “to”) и был ли выбранный курс засчитан (поле “status”).

Trainee

trainee_id	int(10) unsigned	Primary key: unique trainee ID.
name	varchar(255)	Name of trainee.
email	varchar(255)	Email of trainee.

Course

course_id	int(10) unsigned	Primary key: unique course ID.
name	varchar(255)	Name of course.
description	varchar(255)	Description of course.

History

history_id	int(10) unsigned	Primary key: unique history ID.
trainee_id	int(10) unsigned	Trainee ID.
course_id	int(10) unsigned	Course ID.
start	int(11)	Timestamp for when course was started.
end	int(11)	Timestamp for when course was finished.
status	tinyint(4)	Whether the course was passed(1) or not(0).

1. создайте базу данных "Training";
2. создайте указанные выше таблицы;
3. заполните их данными таким образом, чтобы информации было достаточно для выполнения требуемых запросов (т.е. они бы возвращали непустой результат);
4. выведите список стажёров (trainee name, trainee email, course name, history end) которые успешно прошли курс PHP (course name=PHP) в прошлом месяце (сравнить history end);
5. для каждого из всех возможных курсов выведите количество успешных прохождений (course name, quantity) в текущем месяце (сравнить history start);
6. пришлите нам [SQL](#) команды, которыми вы решили эти задания (не просто названия, а полностью со всеми аргументами и параметрами).

Требования к оформлению

- результат выполнения заданий должен быть прислан в виде .zip архива;
- архив должен содержать 5 директорий, по одной на каждое задание, именем директории является номер задания (1, 2, 3, 4 и 5);
- файлы, которые являются точками входа, должны называться *index.html* или *index.php* (при наличии исполняемого PHP кода);
- результатом выполнения задания 5. должен быть один файл с расширением .sql (например, db.sql);
- убедитесь, что все файлы имеют кодировку UTF-8 (не Windows-1251 и пр.);
- при использовании изображений или подключении библиотек, расположенных локально, убедитесь, что они также содержатся в архиве;
- избегайте кириллицы и пробелов в именах файлов;
- в корневую директорию можно (но не обязательно) добавить index.html файл, содержащий ссылки на все выполненные 5 заданий (в задании 5. ссылка на .sql файл).