

Тестовое задание от 09.2020 на позицию бэкенд разработчика (PHP)

В этом тестовом задании Ваша задача разработать небольшой админ интерфейс для каталога сотрудников.

(!) Общая информация

1. Тестовое задание должно быть выполнено с использованием фреймворка **Laravel** и пакета для построения админ интерфейсов **AdminLTE**. Тестовое задание разделено для удобства кандидатов на пункты. Порядок подзадач отражает рекомендуемую последовательность работы над заданием, все подзадачи обязательны к выполнению.
2. Высылайте выполненное тестового задание на электронный адрес **hr@abz.agency**:
 - a. Высылайте решение тестового задания в виде ссылки на github/bitbucket репозиторий.
 - b. Разверните Вашу работу на любом хостинге (чтобы ее можно было посмотреть в браузере не разворачивая проект из исходного кода) и вышлите нам ссылку.
 - c. Укажите в письме список выполненных пунктов и выбранные вами инструменты. Если Вы сделали не все пункты тестового задания – пожалуйста, укажите причину по которой Вы их не выполнили (не хватило времени, не хватает опыта/знания, что-то ещё).
 - d. Обязательно приложите к письму максимально подробное и актуальное на дату отправки резюме.
 - e. Укажите сколько часов было потрачено на выполнение задания.
3. Все материалы, используемые в данном тестовом задании **было специально разработаны** сотрудниками компании abz.agency для проверки знаний соискателей в рамках тестового задания для данной вакансии (wireframes) и не имеют ничего общего с нашими коммерческими проектами. Ваши работы не будут использованы в коммерческих и (даже) некоммерческих целях. Вы также **не имеете права** использовать текст и материалы данного тестового задания, а также Ваше решение данного задания в коммерческих и некоммерческих целях (в том числе в портфолио, для составления курсов или в качестве тестового задания в другой компании).

Важно! Тестовое задание будет принято к рассмотрению только в случае если оно соблюдает требования, перечисленные выше.

(Т) Технические требования

- Laravel 6 LTS / 8+
- [AdminLTE](#)
- [Laravel DataTables](#)
- MySQL
- PHP 7.4+

Важно! При выполнении тестового задания Вы можете дополнительно использовать любые сторонние PHP и/или Javascript/CSS библиотеки, без всяких ограничений. Все 3rd party PHP/Javascript/CSS библиотеки должны быть добавлены в проект через *composer/bower/npm/yarn* если библиотека поддерживает такой способ установки.

(<) Задание

Каталог сотрудников представляет собой базу данных с админ интерфейсом доступным только авторизованным пользователям. По этой [ссылке](#) находятся wireframes админ интерфейса – это то, что должно у Вас получиться после выполнения всех подзадач. При создании UI используйте только готовые компоненты AdminLTE в соответствии с wireframes.

1. Информация о каждом сотруднике должна иметь следующие данные:
 - a. ФИО;
 - b. Должность;
 - c. Дата приема на работу;
 - d. Номер телефона;
 - e. Электронная почта;
 - f. Размер заработной платы;
 - g. Фотография.
 - h. Системные поля: `created_at`, `updated_at`, `admin_created_id`, `admin_updated_id` которые не должны быть доступны для редактирования админу но должны обновляться системой во время CRUD операций.

2. У каждого сотрудника есть ровно 1 начальник. Для простоты будем считать что подчинение одного сотрудника другому не зависит от должности каждого из них, т.е. Два сотрудника могут иметь одинаковую должность но при этом один является начальником другого.
3. База данных должна содержать не менее 50 000 сотрудников и 5 уровней подчинения (в предполагаемой компании не может быть руководителя 6 го уровня).
4. Для должности сотрудника должна быть предусмотрена справочная таблица. Не забудьте добавить системные поля и в эту таблицу.

(←) Подзадачи

1. Создайте базу данных используя миграции Laravel.
2. Используйте Laravel seeder для заполнения базы данных фейковой информацией. Данные сотрудника должны выглядеть реалистично, а не быть просто строкой случайных символов. Мы рекомендуем использовать библиотеку Faker для этого, но можно воспользоваться любой другой с которой Вы больше знакомы.
3. Используя стандартные функции Laravel реализуйте функционал по аутентификации пользователя.
4. Создайте страницу на которой будет выводиться список сотрудников в табличном виде (используя datatables) с отображением всех полей сотрудника, возможностью сортировать по любому полю и осуществлять поиск.
5. Реализуйте остальные CRUD операции для записей сотрудников. После сохранения отправлять пользователя на страницу со списком. Все поля должны быть редактируемыми, включая поле начальника каждого сотрудника. Для всех полей должна быть реализована валидация:
 - а. ФИО: минимум 2, максимум 256 символов;
 - б. Дата приема на работу: валидная дата, формат при выводе дд.мм.гг;
 - в. Номер телефона: валидный номер телефона в международном формате, разрешены только украинские номера;
 - г. Электронная почта: валидный адрес электронной почты;
 - д. Размер заработной платы: дробное число от 0 до 500,000;

- f. Руководитель: учитывать условия для иерархии и должна иметь проверку на целостность данных при сохранении;
 - g. Должность: должна иметь проверку на целостность данных при сохранении;
- 6. Осуществите возможность загружать фотографию сотрудника и отобразите ее на странице, где можно редактировать данные о сотруднике. Добавьте дополнительную колонку с уменьшенной фотографией сотрудника на странице списка всех сотрудников. Ограничения к файлу с фотографией: максимум 5 Мб, минимум 300x300 пикселей, форматы: jpg/png. Выходной формат 300x300px, в jpg (80% quality) вырезаем центр/центр, делаем автоповорот фото при необходимости.
- 7. Осуществите операцию по переподчинению сотрудников в случае удаления записи их начальника. Вы получите дополнительный бонус если реализуете этот функционал с применением встроенных механизмов/парадигм, предлагаемых Eloquent ORM.
- 8. Реализуйте аналогичный CRUD интерфейс для справочной таблицы должностей. Ограничение на длину названия должности: максимум 256 символов. После сохранения отправлять пользователя на страницу со списком.

Важно! Мы рассмотрим каждое решение тестового задания если оно соответствует описанным выше требованиям. Рассмотрение занимает некоторое время, поэтому **не стоит звонить нам на следующий день**, чтобы узнать результат. Так же не стоит спешить и отправлять только часть выполненных подзадач, **только в случае выполнения всех подзадач** ваше решение будет гарантированно рассмотрено нами. У нас открыто несколько вакансий и мы всегда в поиске талантов, проявите себя! Мы желаем Вам удачи и с нетерпением ждем Ваше выполненное задание.