



Программист PHP,JS

Сдача тестового задания:

1. Срок выполнения - **3 суток** с момента получения.

2. Необходимо предоставить:

- **ссылку на репозиторий** (git/bitbucket) с публичным доступом, при разработке отдельно коммитить каждый новый функционал;
- **ссылку на домен** с работающим решением;
- **логин и пароль** от аккаунта с ролью Менеджер;

Проект можно развернуть на **тестовом хостинге**, например, [Beget](#) ([полезные материалы](#)) с бесплатным пробным периодом и техническим доменом.

Требования к выполнению задания:

1. Реализовать задание на фреймворке **Laravel**.
2. При необходимости допустимо использование сторонних JS библиотек.
3. **Чистый** код.
4. Желательно сделать хотя бы частичное **покрытие тестами**.

Задание - Форма обратной связи на Laravel

Необходимо реализовать

1. Регистрация + Авторизация

- 1.1. Стандартный модуль auth, но пользователи должны быть с двумя ролями: Менеджер и Клиент.
- 1.2. Клиенты регистрируются самостоятельно, а аккаунт Менеджера должен быть создан заранее.
- 1.3. После авторизации Клиент видит форму обратной связи (см. п.2), а Менеджер список заявок на обратную связь (см. п. 3).
- 1.4. Все страницы и функционал доступны только авторизованным пользователям в соответствии с их правами.



Программист PHP,JS

2. Страница с формой обратной связи

2.1. Поля формы:

- для ввода темы сообщения;
- для ввода текста сообщения;
- для загрузки файла;
- кнопка для отправки формы.

2.1.1. Размер загружаемого файла не должен превышать 3 МВ.

2.1.2. Не позволять загружать файл с расширениями .bat, .jar, .exe.

2.1.3. Заполнение всех полей обязательно, валидацию реализовать на клиенте и на сервере.

2.2. Клиент может оставлять заявку не чаще раза в сутки.

2.3. В момент обработки формы отправлять менеджеру E-mail со всеми данными.

Отправку почты реализовать асинхронно, используя очереди.

3. Страница со списком заявок на обратную связь

3.1. Поля таблицы:

- ID Клиента;
- Время создания Клиента;
- Время отправки сообщения;
- Имя Клиента;
- E-mail Клиента;
- Тема сообщения;
- Текст сообщения;
- Ссылка на прикрепленный файл.

3.2. В таблице должна быть доступна сортировка по полю "Время отправки сообщения".

3.3. Реализовать пагинацию, с возможностью выбора количества отображаемых строк на странице (10, 50, 100).



Программист PHP,JS

4. Наполнение таблицы данными

4.1. Используя Фабрику, выполнить заполнение таблицы с заявками не менее 500 записями.