

Тестовое задание на позицию Junior Python Developer

В этом тестовом задании вам необходимо разработать платформу для приема заявок на оплату

Общая информация

1. Тестовое задание разделено на две части. Часть No1 обязательная и мы примем Вашу кандидатуру к рассмотрению только в том случае если часть No1 выполнена. Часть No2 опциональная, по количеству задач из этой части которые Вы смогли выполнить мы сможем оценить Ваш уровень опыта и навыков в области веб разработки. Если у Вас уже есть опыт работы с фреймворками Django и/или Flask – попробуйте выполнить все или как можно больше задач из части No2, это повысит ваши шансы по отношению к другим кандидатам.
2. Вы должны отправить Ваше решение для тестового задания нашему HR +7 _____ электронную почту в виде ссылки на github/bitbucket репозиторий. Если Вы пришлете Ваше решение в любом другом виде (в виде ссылки на zip архив, прикрепите zip архив к письму, и др.) – Ваша кандидатура не будет нами рассмотрена!
3. Вам необходимо также приложить ссылку или вложением подробное резюме, контактный номер телефона и ник скайпа.
4. Укажите в письме список выполненных пунктов для каждой части, выбранный вами фреймворк. 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).

Технические требования

- Django 2.1+ / Flask 1.0+
- PostgreSQL 10+
- Python 3.7+

При выполнении тестового задания Вы можете дополнительно использовать любые сторонние Python и/или Javascript/CSS библиотеки, без всяких ограничений. Все 3rd party Python/Javascript/CSS библиотеки должны быть добавлены в проект через pip/bower/npm/yarn если библиотека поддерживает такой способ установки. У нас нет никаких требований к оформлению фронтенд части, но аккуратный вид приветствуется и добавим вам бонусных пунктов.

Часть No1 (обязательная)

Создайте веб страницу, которая будет выводить список заявок на оплату.

- Информация о каждой заявке должна храниться в базе данных и содержать следующие данные:

- ID заявки;
- Сумма;
- Реквизиты;
- Статус (Ожидает оплаты, Оплачена, Отменена)
 - Реквизиты должны быть отдельной таблицей, содержащей следующую информацию: тип платежа (карта, платежный счет), тип карты/счета, ФИО владельца, номер телефона, лимит
 - База данных должна содержать не менее 5000 заявок и 100 реквизитов.

Часть No2 (опциональная)

1. Создайте базу данных используя миграции Django / Flask.
2. Используйте DB seeder для Django ORM / Flask-SQLAlchemy для заполнения базы данных.
3. Используйте Twitter Bootstrap для создания базовых стилей Вашей страницы.
4. Создайте еще одну страницу и выведите на ней список реквизитов со всей имеющейся о них информацией из базы данных и возможностью сортировать по любому полю.
5. Добавьте возможность поиска реквизитов по любому полю для страницы созданной в задаче 4.
6. Добавьте возможность сортировать (и искать если Вы выполнили задачу No5) по любому полю без перезагрузки страницы, например используя ajax.
7. Используя стандартные функции Django / Flask, осуществите аутентификацию пользователя для раздела веб сайта доступного только для зарегистрированных пользователей (создайте таблицу пользователей содержащую роли - пользователь/администратор).
8. Перенесите функционал разработанный в задачах 4, 5 и 6 (используя ajax запросы) в раздел доступный только для зарегистрированных пользователей. Обычные пользователи видят только список реквизитов, администраторы все таблицы
9. Реализуйте с помощью FlaskAPI/Django REST framework API методы:
 - create_invoice (на входе тип реквизитов и сумма, на выходе id заявки и реквизиты)
 - get_invoice_status (на входе id заявки, на выходе статус заявки)
10. С помощью Swagger или Django REST framework создайте описание методов из задачи 9

Пожалуйста не забудьте, что Ваше тестовое задание должно быть предоставлено в виде ссылки на github/bitbucket репозиторий. Мы рассмотрим каждое задание соответствующее заявленным выше требованиям. Рассмотрение занимает некоторое время, поэтому не стоит перезванивать нам на следующий день, чтобы узнать результат. Так же не стоит спешить и отправлять только часть выполненных заданий, если чувствуете что можете сделать больше. У нас открыто несколько вакансий и мы всегда в поиске талантов, проявите себя! Мы желаем Вам удачи и с нетерпением ждем Ваше выполненное задание.