# Тестовое задание

на вакансию junior-yii2-разработчика

Выполнение задания подразумевает изучение документации Yii2 и методичная постановка и выполнение задач для создания своего блога. Этот процесс может занимать от нескольких дней до недель или месяцев, если ты вдруг увлечешься и решишь запрогать убийцу Wordpress. Отнесись к проекту внимательно. Если ты самостоятельно сделаешь это - считай прошел курс молодого бойца и заодно основную фазу испытательного срока.

После ревью делаем персональный оффер и ждем тебя в офисе или на удаленке.

Начни с поиска в Google документации. По Yii2 рекомендуем официальную документацию и украинский сайт с более детальным описанием работы компонентов Yii2.

Далее изучаешь официальный CodeStyle и некоторые соседние документы по ссылке <https://github.com/yiisoft/yii2/blob/master/docs/internals/core-code-style.md>

Разворачиваешь web-сервер (можно встроенный в php) и за дело.

## Стек технологий

* Yii2
* PHP 7.2
* Composer
* SQLLite
* Bootstrap4
* PHPDoc
* PSR cтандарты

## Первый этап

1. Завести репозиторий на gitlab.com
2. Создать проект на Yii2 Basic template и настроить БД
3. Загрузить проект в репозиторий
4. Создать таблицы для блога (записи, категории) через миграции
5. Реализовать функционал создания записей и категорий (одна запись - несколько категорий)
6. Загрузить коммит в репозиторий
7. Реализовать функционал отображения записей (общая лента + по категориям)
8. Коммит в репозиторий

## Требования

* использовать миграции
* использовать ЧПУ (routes/slug)
* сгенерировать CRUD из Gii
* Отображение списка категорий с помощью виджета
* Листинг постов ListView + Pjax
* Вся статика упакована в Assets/Bundle
* Для моделей использовать поведения (Timestamp/Sluggable)

## Второй этап

1. Написать парсер для автоматического наполнения контентом из любого источника (rss/социалка)
2. Запуск по крону и добавляет новые записи в блог.
3. Отправка на почту уведомления о добавлении новых постов
4. REST API для модели с записями блога

##  Требования

* Guzzle Http
* SwiftMailer
* Yii2 Queue

## Результат

Рабочий проект, база собирается из миграций, инициализируется проект, запускается парсер и делается начальная загрузка контента.

При оценке результатов будем смотреть на:

1. соответствие заданию,
2. стиль кода,
3. документирование,
4. ООП,
5. описание коммитов и работа с Git,
6. внешний вид frontend и backend приложений
7. удобство использования

## Комиссия

Принимать решение будем мы: <https://poravinternet.ru/about>

* Степан Чельцов, General Director
* Дмитрий Половков, Backend Developer
* Ярослав Старушенко, Frontend Developer

### Мы желаем успехов и креативного настроения!

###