**Test Task**

**Fullstack Web Developer**

**Требования к Разработке**

1. Общие требования
   1. Разработать Главную страницу на **CMS WordPress**.
   2. Использовать последнюю версию CMS WordPress.
   3. Скачать и установить **английскую версию** с английского сайта <https://wordpress.org>.
   4. Разработку производить с помощью [**Child Темы**](https://codex.wordpress.org/%D0%94%D0%BE%D1%87%D0%B5%D1%80%D0%BD%D0%B8%D0%B5_%D1%82%D0%B5%D0%BC%D1%8B) к любой стандартной теме поставляемой с установкой **CMS WordPress**.
   5. Каждый пункт меню должен вести на свою страницу (страницы стилить не нужно).
   6. Учитывать что разработка делается для клиента, который не знает языков программирования таких как **HTML**, но при этом будет хотеть менять контент.
   7. Размещение функциональных элементов в Dashboard должна быть **интуитивно понятно**.
   8. Разработка должна легко переносится с сервера на сервер с **минимальным** набором запросов в базу.
   9. Кроме Главной страницы ничего более разрабатывать **не нужно** (внутренние страницы, портфолио, блог и тд.).
   10. Комментарии в коде должны быть на **английском**.
2. Содержание папки с Тестовым Заданием
   1. Папка **PNGs** - Папка с изображениями дизайна.
      1. **Home.png** - Дизайн Главной страницы.
      2. **Hover.png** - Раскадровка с ховерамии выпадающим меню.
      3. **Structure.png** - Структура страницы.
      4. **Styleguide.png** - Руководство по стилю и цветовая схема.
   2. Папка **PSDs** - Папка с дизайнами.
      1. **Home.psd** - Дизайн Главной страницы.
      2. **Styleguide.psd** - Руководство по стилю и цветовая схема.
   3. **Test Task Fullstack Web Developer** - Техническое Задание
3. Требования к Front-End
   1. Дизайн Главной страницы должен **соответствовать** Home.psd.
   2. Раскадровка с ховерамии выпадающим меню находится в Hover.png.
   3. Допускается использование [Bootstrap Framework](https://getbootstrap.com) или подобного.
   4. **Запрещено** использовать конструкторы такие как Visual Composer или подобные, кроме встроенного Gutenberg.
4. Требования к адаптивности дизайна
   1. Дизайн должен адаптироваться под **все** разрешения. То есть быть Liquid.
   2. Дизайн должен выглядеть одинаково в браузерах[**Chrome**](https://www.google.com/chrome/), [**Firefox**](https://www.mozilla.org/).
   3. Не должно быть **горизонтальных полос** прокруток.
5. Требования к Back-End
   1. Пермалинки должны быть вида **Post Name**.
   2. **Front Page** должен быть назначен.
   3. Разработка должна производиться на **PHP 5.6+** или **PHP 7+**.
   4. Допускается использование Advanced Custom Fields плагина для отдельных полей. Запрещается полностью строить на нем страницу.
   5. PHP скрипты **должны** находится в **functions.php** или стороннем файле функций.
   6. Подключение [скриптов](https://developer.wordpress.org/reference/functions/wp_enqueue_script/) и [стилей](https://developer.wordpress.org/reference/functions/wp_enqueue_style/) front-end **должно** производится в **functions.php**.
   7. Главная страница должна **находится** в темплейте [**front-page.php**](https://codex.wordpress.org/%D0%98%D0%B5%D1%80%D0%B0%D1%80%D1%85%D0%B8%D1%8F_%D1%88%D0%B0%D0%B1%D0%BB%D0%BE%D0%BD%D0%BE%D0%B2).
   8. Для слайдера можно использовать плагин, но рекомендуется любой скриптовый js слайдер и создать его на основе Post Type.
6. Требования к Плагинам
   1. Разработка должна быть максимально проведена нативно. **Использование плагинов должно быть обоснованно**.
   2. **Запрещенные плагины**
      1. Visual Composer или подобные.
      2. Custom Post Type или подобные.
      3. Платные или сторонние темы.
   3. **Разрешенные плагины**
      1. Advanced Custom Fields и подобные.
      2. Плагины форм.
      3. Плагины Слайдеров и Каруселей.
7. Требования к Структуре
   1. Структура разбита на блоки и показана в файле **Structure.png**.
   2. Основные структурные элементы
      1. **1** - Выпадающее меню.
      2. **2** - Меню являющееся Main Menu CMS WordPress.
      3. **3** - Слайдер. Изображения. Заголовок и текст каждого слайда должны меняться из Dashboard.
      4. **4** - Кастомные поля. В каждом из блоков - иконка, заголовок и текст должны меняться из Dashboard.
      5. **5** - Карусель Portfolio. Должен быть состоять из сингл постов Post Type. Все посты должны иметь ссылку на сингл страницу и находится в карусели. [Post Type](https://codex.wordpress.org/Function_Reference/register_post_type) должен быть создан в functions.php.
      6. **6** - Title страницы. H1. Должен меняться из Dashboard.
      7. **7** - Вывод из [the\_content()](https://developer.wordpress.org/reference/functions/the_content/). Должен меняться из Dashboard.
      8. **8** - Стандартный виджет Меню. Меню идентичное главному. Должен меняться из Dashboard.
      9. **9** - Стандартный текстовый виджет. Должен меняться из Dashboard.
      10. **10** - Форма отправки. Должна работать. Рекомендуется использовать [Contact Form 7](https://wordpress.org/plugins/contact-form-7/).
      11. **11** - Копирайт. Должен меняться из Dashboard.
8. Требование к возврату Задания
   * 1. После завершения выполнения тестового задания его необходимо упаковать в **ZIP** архив и выслать через скайп, на почту или предоставить ссылку на скачивание.
     2. В состав архива должны входить:
        1. **Вся** папка CMS WordPress.
        2. База данных в формате **SQL**.
        3. Пояснительная записка в произвольной форме с указанием использованной версии **PHP** и комментариями и **доступами** в Dashboard.
9. Как мы проверяем Тестовое задание
   * 1. Мы смотрим на сколько полон архив и как легко его удается поднять у нас на сервере.
     2. Длительность выполнения. Чем дольше тем более детально.
     3. Задание должно максимально соответствовать дизайну. В идеале [PixelPerfect](https://www.unisender.com/ru/support/about/glossary/chto-takoe-pixel-perfect-verstka/), но это не обязательно.
     4. [PageSpeed Insights](https://developers.google.com/speed/pagespeed/insights/) должен составлять 55+/65+.
     5. [W3C Markup Validator](https://validator.w3.org) должен содержать только предупреждения.
     6. Соответствие всем пунктам требований описанных выше. Они все важны так как взяты на основе реальных проектов.
     7. Мелкие детали такие как ховеры, цвета, шрифты и тд.
     8. Респонсив проверяется изменением ширины браузера, далее на реальных устройствах.
     9. Смотрится код на наличие бекенд и php скриптов.
     10. Смотрится код на наличие хардкода (недоступного для редактирования из Dashboard текста) и копипаста.
     11. Проверяется наличие ошибок PHP при включенном дебагере.
10. Правила выполнения Задания
    * 1. Задание рассчитано на 6-8 часов работы.
      2. Время на выполнения задания 3-4 календарных дня.
      3. Задание стоит брать в день или за день до фактического начала выполнения.
      4. Любые навыки разработки сделанные сверх задания приветствуются.
      5. **Не нужно отправлять незаконченные или промежуточные варианты задания.**
      6. **Не нужно заливать задание на собственный сервер. Мы его поднимаем у себя.**
      7. Если допущены незначительные ошибки или погрешности, они будут разобраны с Вами при заходе на триал.

**-------------------------------------------------------------------------------------------------**

Это задание позволит определить Ваш уровень и внимательность к деталям, что требуется при выполнении реальных проектов.

Мы не будем строго подходить к оценке задания. Мы хотим увидеть на что вы способны и чем мы Вам сможем помочь в нашей команде.