**Общие навыки**

1. Базовое знание английского языка

2. Умение находить информацию, гуглить

3. Работа с Git. Github Desktop, Gitlab, PuTTy, FileZilla, Composer

4. Поверхностные знания Photoshop, Figma, Illustrator

**Клиентская разработка**

1. HTML

2. JavaScript

3. CSS

1. Использование препроцессоров, вместо CSS

Самые популярные: SASS, LESS, Stylus, PostCSS

2. Использование фреймворков

Vue 3, Nuxt 3, React

3. Умение использовать и подключать библиотеки

Swiper, Chart, UI-библиотеки

4. Понятие о шаблонизаторах

5. Использование БЭМ

**Серверная разработка**

**API -** связующее звено между клиентом и сервером

Разработка API на Laravel, Node JS, ExpressJS, NextJS, NestJS

Модули: Sequelize, multer, JWT

<https://sequelize.org/>

<https://expressjs.com/ru/>

понимание ООП, MVC (Model, View, Controller), паттерны

PHP 7.4+

Laravel

**База данных**

MySQL, SQLServer

Понимание запросов SQL на продвинутом уровне, представления, функции, триггеры, события, процедуры, индексация, пользователи БД, права, конфигурация БД, связи.

Для управления БД PHPMyAdmin, SSMS

**Системы управления контентом**

WordPress

Joomla

Drupal 7+

OpenCart

ModX

Magento

**Конструкторы**

Tilda

**Github Desktop**

Github Desktop - это инструмент для удобной работы с репозиториями на Github. Ниже я расскажу про основные команды.

1. Clone - создание локальной копии репозитория на компьютере. Для этого нужно выбрать в меню File -> Clone Repository и ввести адрес репозитория.

2. Pull - скачивание изменений с удаленного репозитория. Для этого нужно нажать на кнопку Pull в верхней части окна приложения. Это может быть нужно, если другой участник проекта внес изменения в репозиторий.

3. Push - загрузка изменений на удаленный репозиторий. Для этого нужно нажать на кнопку Push в верхней части окна приложения. Это может быть нужно, если вы сделали изменения в локальной копии репозитория.

4. Fetch - скачивание информации о состоянии удаленного репозитория без скачивания его изменений. Для этого нужно нажать на кнопку Fetch в верхней части окна приложения.

5. Branches - создание, изменение и удаление веток. Ветки позволяют разрабатывать различные варианты приложения в рамках одного репозитория. Для этого нужно выбрать в меню Branch -> New Branch и ввести имя новой ветки.

6. Commit - сохранение изменений в локальной копии репозитория. Для этого нужно написать комментарий и нажать на кнопку Commit в нижней части окна приложения.

7. Merge - объединение изменений из разных веток. Для этого нужно выбрать в меню Branch -> Merge into Current Branch и выбрать нужную ветку.

**Roadmap`ы по изучению технологий.**

Vue, Nuxt - обязательно Composition API

HTML - обязательно методология БЭМ

CSS - обязательно SCSS

Обязательно расширение Prettier (чтобы у каждого были одинаковые настройки)



**PHP**

**Уровень 1: Основы**

- Установка и настройка окружения для работы с PHP

- Основные принципы языка: переменные, операторы, условные конструкции, циклы, функции

- Работа с базами данных, включая SQL запросы с помощью PHP

- Работа с формами и отправка данных на сервер

- Создание и работа со строками, массивами и объектами

- Ошибки и исключения

**Уровень 2: Продвинутый**

- ООП в PHP: классы, объекты, наследование, полиморфизм

- Работа с различными типами данных: JSON, XML и другие форматы

- Работа с файлами: чтение и запись, загрузка файлов на сервер

- Работа с URL и параметрами

- Работа с сессиями и куками

- Работа со строками и регулярными выражениями

**Уровень 3: Расширенный**

- Продвинутые концепции ООП: интерфейсы, абстрактные классы, трейты

- Работа с расширениями и библиотеками: Composer, PHPUnit, Guzzle, Symfony и др.

- Работа с асинхронным кодом: многопоточность, параллельное выполнение, асинхронные запросы

- Работа с кэшем и оптимизация производительности

- Работа с фреймворками: Laravel, CodeIgniter, Yii, Symfony и др.



**Уровень 4: Экспертный**

- Безопасность в PHP: защита от SQL инъекций, XSS атак, CSRF атак и других угроз

- Работа с архитектурой приложения: паттерны проектирования, SOLID принципы, MVC архитектура

- Построение и работа с API: REST, SOAP, GraphQL

- Оптимизация производительности: оптимизация кода, оптимизация SQL запросов, масштабирование приложения

- Дебаггинг и тестирование: использование дебаггеров, автоматическое тестирование с помощью PHPUnit и других инструментов.

**Вёрстка**

**Уровень 1: Основы**

- HTML и CSS: изучение синтаксиса и базовых конструкций, создание простейших вёрсток.

- Кроссбраузерность и валидность: разбор особенностей вёрстки под различные браузеры и проверка валидности HTML и CSS кода.

- Респонсивность: создание адаптивных вёрсток под различные устройства.

- Как правильно верстать? Часть 1.

<https://habr.com/ru/articles/655009/>

**Уровень 2: Продвинутый уровень**

- Как правильно верстать? Часть 2.

<https://habr.com/ru/articles/664060/>

- Advanced CSS: изучение деталей работы с CSS, специфичных свойств, анимации и эффектов.

- Шаблоны и фреймворки: использование готовых шаблонов и фреймворков для быстрого создания вёрсток.

- Препроцессоры CSS: изучение Sass или Less, создание переменных, миксинов и функций для более удобного и быстрого написания CSS.

**Уровень 3: Расширенный уровень**

- Работа с JavaScript: базовые знания, работа с DOM и программирование анимаций.

- Интеграция различных технологий: настройка сборщиков проектов и выпуск страниц на сервер с применением различных технологий.

- SVG и анимации: работа со векторной графикой и создание анимаций на основе SVG.

**Уровень 4: Экспертный уровень**

- CSS-архитектура: понимание принципов работ Twitter Bootstrap и других крупных библиотек CSS, организация кода с помощью БЭМ, методологии SMACSS.

- Работа с архитектурой sitecore или других популярных CMS платформ: создание каталогов и других разделов сайта в соответствии с требованиями.

- Разработка собственных компонентов: создание собственных компонентов, переиспользование кода и улучшение структуры проекта.

**JavaScript**

**Уровень 1: Основы**

- Синтаксис и основные концепции JavaScript

- Типы данных и переменные

- Операторы и выражения

- Управляющие конструкции (условные операторы, циклы)

- Функции и области видимости

**Уровень 2: Продвинутый**

- Объекты и массивы

- Работа с DOM

- События и обработчики

- FETCH и работа с сервером

- ООП в JavaScript

**Уровень 3: Расширенный**

- Паттерны проектирования в JavaScript

- Фреймворки и библиотеки (React, Angular, Vue)

- Тестирование и отладка

- Оптимизация производительности

- Работа с WebSockets и фоновыми задачами

**Уровень 4: Экспертный**

- Разработка архитектуры больших приложений

- Серверный JavaScript (Node.js)

- Применение WebAssembly

- Работа с криптографией

- Создание собственных библиотек и фреймворков

* **Фреймворки**

**Laravel**

Этот роадмап по изучению Laravel предоставляет общую структуру и рекомендации по тому, что вы должны изучить и какие навыки получить для эффективного использования фреймворка Laravel.

**Уровень 1: Основы**

**PHP**

- Синтаксис PHP

<https://www.php.net/manual/ru/language.basic-syntax.php> (справа темы)

- Основные типы данных PHP

<https://www.php.net/manual/ru/language.types.php> (справа темы)

- Условия и циклы

<https://www.php.net/manual/ru/control-structures.intro.php> (справа темы)



**Основы веб-разработки**

- HTTP, HTTPS

<https://youtu.be/2IMaWdUXkqE>

- Использование HTML, CSS и JavaScript на клиентской стороне

- Как работает серверное программирование

<https://youtu.be/mus6OwoJ90I> (видео на 2 темы)

**Основы баз данных**

- Язык SQL

<https://youtu.be/GQfC0nYrto8>

<https://youtu.be/pSg8qlMXKf4>

- Нормализация баз данных

<https://youtu.be/zqQxWdTpSIA>

* **Laravel**

- Установка и настройка Laravel

- Понимание структуры каталогов Laravel

- Работа с роутами

- Работа с контроллерами

- Основы работы с Eloquent ORM

**Уровень 2: Продвинутый уровень**

**PHP**

- ООП в PHP

- Исключения и обработка ошибок

* **Laravel**

- Работа с Blade

- Работа с миграциями и фасадами

- Работа с запросами Eloquent

- Middlewares

- Основы работы с PHPUnit

**Уровень 3: Расширенный уровень**

**PHP**

- PHP и PSR стандарты

- Кеширование в PHP

- Расширение PHP с помощью модулей, библиотек и расширений



**Laravel**

- Конфигурирование Laravel

- Работа с событиями и очередями в Laravel

- Работа со множественными базами данных

- Оптимизация Laravel

- Использование Redis и Memcached с Laravel

**Уровень 4: Экспертный уровень**

Экспертный уровень разработки с Laravel предназначен для программистов, которые уже имеют значительный опыт работы с фреймворком и хотят продвинуться на новый уровень.

**PHP**

- Углубленное изучение ООП в PHP

- Рефакторинг кода

- Шаблонные методы и паттерны проектирования



**Laravel**

- Создание пользовательских фасадов и пакетов Laravel

- Laravel Horizon для управления очередями

- Работа с RabbitMQ в Laravel

- Работа с мульти-tenancy в Laravel.



**Nuxt 3**

Nuxt 3 - это следующее поколение популярного фреймворка Nuxt.js для создания универсальных приложений на Vue.js. Он будет предлагать множество новых возможностей, таких как улучшенная производительность, более интуитивный API и многопоточность.

**Уровень 1: Основы**

- Знакомство с Nuxt.js и его основными концепциями

- Создание простого приложения с помощью Nuxt.js

- Работа с маршрутизацией и параметрами маршрутов в Nuxt.js

- Работа с шаблонизацией и компонентами в Nuxt.js

- Использование Pinia для управления состоянием приложения

- Использование встроенным composable в Nuxt 3 - fetch для обращения к API

**Уровень 2: Продвинутый**

- Оптимизация загрузки приложения с помощью различных методов, таких как prerendering, prefetching и lazy loading

- Работа с серверной рендерингом и статической генерацией

- Использование middleware для обработки запросов на сервере перед рендерингом страницы

- Использование layout и его настройка

- Использование plugins для добавления функциональности к приложению

- Использование env переменных для настройки конфигурации приложения

**Уровень 3: Расширенный**

- Создание собственных плагинов и модулей для Nuxt.js

- Использование TypeScript для разработки приложения

- Работа с библиотеками для управления состоянием, такими как Redux и MobX

- Использование GraphQL для запросов к API

- Использование изоморфных библиотек для работы с базами данных и другими backend-технологиями

**Парсинг**

**Язык:** Python

<https://docs.python.org/3/>

**Roadmap Python**

**Уровень 1: Основы**

- Базовый синтаксис

- Переменные и типы данных

- Списки, словари, кортежи, множества

- Приведение типов, исключения

- Циклы и условия

- Функции, встроенные функции

- Виртуальное окружение

**Уровень 2: Продвинутый уровень**

- Классы и ООП

- Инкапсуляция, наследование, полиморфизм

- Методы

- Декораторы

- Итераторы и генераторы

- RegEx - регулярные выражения

- Структуры данных и алгоритмы:

- Массивы и связанные списки

- Кучи, стеки и очереди

- Хэш-таблицы

- Бинарные деревья

- Рекурсия

- Алгоритмы сортировки

**Библиотеки:**

**Request**

<https://requests.readthedocs.io/en/latest/index.html>

https://python-scripts.com/requests

**BS4**

<https://beautiful-soup-4.readthedocs.io/en/latest/>

<https://www.youtube.com/@PythonToday>

**Selenium**

<https://selenium-python.readthedocs.io>

https://questu.ru/articles/81673/

<https://www.youtube.com/playlist?list=PL6plRXMq5RABXsb0-B0CrWniWBoFZOL20>

**Pymysql**

<https://pymysql.readthedocs.io/en/latest/index.html>

https://absolutecodeworks.com/python-pymysql-tutorial

**Фреймворки:**

**Scrapy**

<https://docs.scrapy.org/en/latest/index.html>

<https://www.scraperapi.com/blog/scrapy-web-scraping/>

https://www.youtube.com/watch?v=n6pq0RjfpEM