

Программа обучения BACKEND BOOTCAMP

[PHP]Пр

- [OOP] Введение в ООП
- [OOP] Паттерн MVC (Model-View-Controller)
- [OOP] Паттерны Data Mapper и Active Record
- [OOP] Внедрение зависимостей (Dependency Injection)

- [Linux] Работа с Git
 - [Linux] Установка пакетов
 - [Linux] Конфигурация LAMP
 - [Linux] Устранение неисправностей

- [Database] Понятие базы данных
 - [Database] Принцип работы реляционных баз данных
 - [Database] Принципы ACID
 - [Database] Определение таблиц и полей
 - [Database] Создание связей между записями с помощью первичных и внешних ключей
 - [Database] Написание запросов к БД на языке SQL
 - [Database] Работа с базой данных с помощью Workbench/PhpStorm
 - [Database] Отладка запросов с помощью EXPLAIN

- [Git]Rebase
 - [Git]Push/Pull
 - [Git]SSH/HTTP-доступ

 - [Form]Дизайн форм и извлечение данных на сервере
 - [Form]Валидация клиентских данных
 - [Form]Защита данных перед обработкой
фильтрация

- [Composer]Обновления зависимостей
 - [Composer]Установка зависимостей
 - [Composer]Packagist для поиска пакетов

 -  [Filesystem]Создание, запись и удаление файлов и папок
 - [Filesystem]Поиск по файлам и директориям
 - [Filesystem]Сортировка

 -  [Doctrine]OOP подход при работе с БД
 - [Doctrine]Обновление схемы БД

- [Symfony] Управление зависимостями через Composer
 - [Symfony] Бандлы (Bundles) и Паки (pack)
 - [Symfony] Работа с БД на примере ORM Doctrine
 - [Symfony] Работа с шаблонами на примере Twig
 - [Symfony] Symfony Best Practices
 - [Symfony] Symfony Components
 - [Symfony] Маршрутизация в Symfony приложении
 - [Symfony] Формы в Symfony приложениях
 - [Symfony] Кеширование в Symfony приложении
 - [Symfony] Безопасность в Symfony приложении
 - [Symfony] Консольные команды в Symfony

- [Doctrine]Behaviors (Tree, Translatable, Loggable, Sluggable, Timestampable)
 - [Doctrine]ООП подход при работе с БД
 - [Doctrine]Обновление схемы БД
 - [Doctrine]Работа с фикстурами
 - [Doctrine]Сохранение, редактирование, обновление и удаление объектов из БД
 - [Doctrine]Настройка связей между сущностями (OneToMany, ManyToOne, ManyToMany)

- [Twig] Дебаг
 -  [REST API] Конфиг
 - [REST API] Сериализация
 - [REST API] CRUD
 - [REST API] JWT

- The diagram illustrates the relationship between the 'Console' feature and other application components. At the top, there are seven rounded rectangular boxes representing different features: ШАБЛОНИЗАЦИЯ (Template), РОУТИНГ (Routing), ORM, ФРЕЙМ (Frame), КОНСОЛЬ (Console), ЛОГИРОВАНИЕ (Logging), and БЕЗОПАСНОСТЬ (Security). The 'Console' box is highlighted with a blue star icon to its left. Below the diagram, four items are listed under the heading '[Console]': Создание консольных команд (Creating console commands), Стилизация вывода (Styling output), Дебаг (Debug), and Консольные приложения (Console applications).

- [Blackfire] Конфигурация
 - [Blackfire] Профилирование цепочек вызова php-кода и ресурсов (БД, сеть, память, процессор)
 - [Blackfire] Разбор отчетов и графов
 - [Docker] Установка и принцип работы
 - [Docker] Yml config
 - [Docker] Запуск и управление контейнерами
 - [Docker] Docker в контексте GitLab CI
 - [Testing] PHPUnit
 - [Testing] Selenium

- [Security] CSRF защита
 - [Security] Конфигурация

 - [BestPractices] Фильтрация и валидация пользовательских данных
 - [BestPractices] Безопасные SQL запросы для предотвращения SQL-инъекций
 - Установка Composer
 - [BestPractices] Рефакторинг кода для ухода от дублирования (DRY, KISS)
 - [BestPractices] Стандарты кодирования и соглашения (Code Style, PSR)
 - [BestPractices] Contribution в Open Source библиотеки
 - [BestPractices] SOLID/XP/TRIZ

 - [Continuous Integration] Gitlab CI
 - [Continuous Integration] Unit и Functional-тесты

[Testing]

- [Continuous Integration]Code style
- [Continuous Integration]Статическая проверка кода
- [Continuous Integration]Auto-deployment
- [Continuous Integration]Привязка коммитов к таскам

PHP School

6 месяц

- УПАКОВКА ВАШЕГО РЕЗЮМЕ

- [HighLoad]Отказоустойчивость (availability)
 - [HighLoad]Nginx + php-fpm
 - [HighLoad]Нагрузочное тестирование

- [Symfony] Symfony Service Container
 - [Docker] Dockerizing workflow
 - [Docker] Orchestration
 - [Docker] Pull

- [WebSocket]WAMP proto
 - [WebSocket]Swoole

 -  [Практика] Выбор и проектирование
 - [Практика] Реализация бизнес-логи

- [Contributing]Github Social Coding
 - [Contributing]Fork, Pull Request
 - [Contributing]Ваша open source активность
 - [Contributing]Как находить репозитории для contributing
 - [Contributing]Как правильно контрибутизировать оформлять pull request-ы

[Практическое]

Computing]Github Social Coding

Computing]Fork, Pull Request

urce активность репозитории для

- резюме (CV)
 - [Marketing]Поиск и подбор проектов
 - [Marketing]Разбор и прохождение интервью
 - [Marketing]Прохождение исп срока