Backend developer

1. Что делает человек этой роли:
   1. Обеспечение корректной работы всех функций сайта и его вычислительной логики;
   2. Организация и работа с базами данных посредством СУБД;
   3. Разработка базовой логики и алгоритмов работы приложения;
   4. API;
   5. Интеграция с внешними сервисами;
   6. Тестирование и отладка приложения и отдельных компонентов.
2. Инструменты:
   1. Back4app-Сервис для масштабируемого написания BackEnd решений
   2. Firebase-набор инструментов и сервисов для разработки BackEnd решений для Веб и мобильных прложений
   3. Nginx-взаимодействие с сервером
   4. Docker-Контейнеры
   5. Kubernetes-Контейнеры
   6. Jira-Система отслеживания выпусков ПО
   7. Pivotal Tracker-Система отслеживания выпусков ПО
   8. Slack-Коммуникационное программное обеспечение
   9. Google Pagespeed Insights-Производительность веб-сайта
   10. Apache-Программное обеспечение для веб-серверов
   11. Github-Управление версиями
   12. Gitlab-Управление версиями
   13. Bitbucket-Управление версиями
   14. Datadog-Сервис мониторинга
   15. New Relic-Служба мониторинга
   16. Stackoverflow-Сайт вопросов/ответов
   17. Gitkraken-Git GUI клиент для Windows, Mac и Linux
   18. Visual Studio-IDE
   19. Postman API-Платформа совместной работы для разработки API
3. Знания и навыки
   1. Строго типизированный язык Go или Java;
   2. Умение работать с каким-либо из веб-серверов для выбранного языка, например, Echo/Gin для Go или Tomcat/Jetty для Java;
   3. Фреймворки: Nest.js(Node.js), Django(Python), Spring Boot(Java/Kotlin)
   4. Базы данных – SQL (Postgres) и/или нереляционная база данных – MongoDB/DynamoDb
   5. Архитектурные паттерны: Микросервисная архитектура, Git, Jira, Trello
   6. Кэши, например, Redis;
   7. Контейнеризация объектов: Docker, Kubernetes
   8. Очереди, Kafka, SQS;
   9. Облачные провайдеры – Amazon, Azure, Google Cloud;
   10. Метрики, знание как работают и умение работать со стандартными пакетами вроде prometheus;
   11. Принципы и виды тестирования – юнит, компонентные, и интеграционные тесты.
4. Взаимодействия с ролями
   1. Frontend: Работа с API, Реализация логики приложения, Тестирование интеграции
   2. DevOps: Настройка окружения(Запуск в контейнерах, настройка серверов, CI/CD)
   3. С тестировщиком: передача функционала на тестирование, Работа с багами, Написание тестов(юнит- и интеграционные тесты)
5. Зона ответственности
   1. Архитектура и логика приложения
   2. Работа с БД
   3. Создание и поддержка API
   4. Безопасность
   5. Интеграция с внешними сервисами
   6. Тестирование и отладка
   7. Поддержка и масштабирование
6. Отличия от других ролей
   1. FrontEnd: В отличии от него, реализует логику работы программы и результат его работы напрямую невидим пользователю
   2. DevOps: BackEnd пишет сам код, а DevOps отвечает за работу инфраструктуры и запуск этого кода на ней
   3. Тестировщик: BackEnd тестирует только свой код и отвечает за написание самого кода
7. Подходит ли роль: Роль имеет несколько пунктов, которые меня в ней привлекают, такие как реализация математической логики, работа с БД, малое количество субъективности. Из минусов можно отметить высокий уровень ответственности, но плюсы компенсируют это