

Корначук Марк

Город: Москва

GitHub | Email: kornachuk.mark@gmail.com | Telegram: [@potatohd404](https://t.me/potatohd404)

О себе

Я Full Stack разработчик с опытом работы в React, Typescript, C#, Python, SQL и Linux. У меня есть опыт в разработке веб-приложений, управлении базами данных и взаимодействии с API. Я постоянно самосовершенствуюсь и изучаю новые технологии и методологии, что помогает улучшать качество моей работы.

Навыки

Языки	: Python, TypeScript, C#, SQL, Bash
Фреймворки	: React.js, Next.js, ASP.NET, Django
Библиотеки	: Redux, React Router
Базы данных	: PostgreSQL, Firebase, DynamoDB
Инструменты	: Git, GitHub, Terraform, Docker
Облака	: Yandex Cloud, AWS, Vercel

Образование

Национальный Исследовательский Ядерный Университет "МИФИ"	Бакалавриат
Программная инженерия	Сентябрь 2020 – Июль 2024

Проекты

daily-mephi.ru	Next.js, Typescript, Serverless, Yandex cloud	Исходный код
<ul style="list-style-type: none">Разработал студенческий портал для отзывов о преподавателях и обмена материалами с использованием Next.js, TypeScript, Prisma, tRPCПолностью развернут в Yandex cloud при помощи terraform и yandex cloud functions (serverless)В проекте настроен Github Actions для непрерывной разработки и тестированияДля тестирования в проекте используются Jest (для интеграционных и unit-тестов) и Storybook (для тестов компонентов)В ходе проекта был также разработан способ загрузки файлов в notion для замены облачного хранилищаКроме того был разработан ORM для YDB на Node.jsБыла создана система рендеринга React компонентов в PNGБыл создан прототип библиотеки для представления бекенда на next.js в виде, похожем на Spring (в виде классов), а также dependency injection для этой библиотекиГлавным вызовом в проекте было развертывание в облаке, так как не существовало технологии развертывания next.js в yandex cloud		
ParallelDB	C#, ASP.Net, AWS Lambda, Vue3, Typescript	Исходный код
<ul style="list-style-type: none">Простая база данных на C#, позволяющая обрабатывать запросы параллельноНаписан парсер для SQL, DSL для программного использованияAPI на ASP.net для взаимодействия с фронтендом, захосчено на AWS lambdaФронтенд на Vue3 для отправки запросов к БД и последующего отображенияНаибольшую сложность представлял парсер SQL и последующую обработку запроса		
website-checker	Telegram, Golang, AWS lambda, DynamoDB	Исходный код
<ul style="list-style-type: none">Телеграм бот для проверки изменений сайта, написанный на golangКаждые 5 минут происходит проверка изменений всех сайтов из базы данных DynamoDBМожно подписаться на изменения любого сайта, но только админ может добавить новыйОсновной сложностью было настроить terraform, GitHub actions и AWS SDK		