

Иванов Максим Сергеевич

Веб-разработчик



ОБО МНЕ

Веб-разработчик с 4-летним опытом создания сложных многокомпонентных приложений. Специализируюсь на fullstack-разработке, начиная от создания интерфейсов и заканчивая серверной логикой и работой с базами данных. Имею опыт работы как над собственными и групповыми проектами, так и в коммерческих командах, где успешно реализовал проекты различной сложности. Постоянно улучшаю свои навыки и внедряю современные технологии для создания масштабируемых и высокоэффективных веб-приложений.

ОПЫТ РАБОТЫ

Fullstack Веб-разработчик

Фриланс / Пет-проекты:

- Разработка fullstack приложений с использованием React, Next.js и Node.js (также Gin, .NET, FastAPI).
- Разработка UI на основе компонентов с применением React и TypeScript.
- Оптимизация производительности клиентской части, включая рендеринг и загрузку данных.
- Разработка серверной логики и API на Node.js с использованием баз данных (PostgreSQL, MySQL, MongoDB).
- Упаковка и развертывание приложений с использованием Docker локально и на виртуальном сервере (VPS).
- Использование Git.
- Интеграция с внешними API и создание собственного функционала для взаимодействия с базами данных.

Примеры проектов:

- **Приложение для мониторинга данных с датчиков буровых установок и выявления аномалий** (дипломный проект): веб-приложение представляет собой панель для мониторинга, на которой строятся графики по данным, приходящих в режиме реального времени с датчиков буровых установок. В зависимости от выбранного метода

КОНТАКТЫ

+7 (921) 363-04-35
makschocomint@gmail.com
github.com/MaksChocomint
telegram: @makschocomint

КЛЮЧЕВЫЕ НАВЫКИ

Уровень английского: C1

Frontend: HTML, CSS, SASS, TailwindCSS, JavaScript, TypeScript, React, Redux, Zustand, Next.js

Backend: Node.js, Express.js, Prisma ORM, Python (FastAPI, SQLAlchemy), Go (Gin)

Базы данных: PostgreSQL, MySQL, MongoDB

DevOps: Docker

Другие технологии: REST API, GraphQL, WebSockets, Git, Java Spring, C++ (Qt Creator), C# .NET

и настроек, алгоритм отмечает на графиках аномальные значения и уведомляет об этом пользователя. На основе использованных методов был разработан новый универсальный метод обнаружения аномалий на основе контекстного анализа буровых данных.

Технологии: React (NextJS), TypeScript, plotly.js, PostgreSQL, Prisma, WebSockets, Python (FastAPI).

- **FinOps Insight Portal** (пет-проект): fullstack-портал финансовой аналитики и поддержки бизнес-пользователей. Реализованы KPI-дашборды (выручка, ARPU, churn), управление регламентными задачами по SLA, система тикетов на доработки и AI-ассистент для объяснения аномалий, SQL-подсказок и краткого резюмирования заявок. Приложение разворачивается в Docker-контейнере с PostgreSQL, Redis и MinIO. Технологии: React, TypeScript, Zustand, Recharts, Go (Gin), JWT, PostgreSQL, Docker, Redis, MinIO, OpenAI API.

- **Онлайн-игра "Codenames"** (пет-проект): популярная настольная командная игра, где игроки соревнуются друг с другом в придумывании и отгадывании слов-ассоциаций. Реализованы функции создания игровой комнаты, игры в режиме реального времени, доступа к различным режимам игры, недоступным в аналогах.

Технологии: React (NextJS), Redux, JavaScript, Express.js, Node.js, WebSockets (Socket.io).

- **Сайт по бронированию квартир и домов** (заказ): многостраничный сайт с возможностью ознакомления с управляющей компанией, специализирующейся на посуточной аренде домов и квартир. Реализованы функции просмотра доступных апартаментов и их бронирования с оплатой.

Технологии: React (NextJS), TypeScript, Prisma, PostgreSQL.

- **Система генерации аналитического контента минерально-сырьевого комплекса** (курсовой запатентованный проект): изначально был адаптирован буклет "Молодежный прогноз развития энергетики", где изображены графики различного вида. Задача заключалась в создании админ-панели для добавления текстов и графиков, и страниц для просмотра блоков буклета. Приложение масштабировалось в систему по созданию аналитического контента. Была добавлена возможность прогнозирования данных для следующих лет с помощью регрессионной модели/других моделей прогноза данных, которые исследовательская группа университета может сама добавлять.

Технологии: React (NextJS), TypeScript, chart.js, Prisma, PostgreSQL.

Другие примеры можно посмотреть на
github.com/MaksChocomint
Или на моем сайте-портфолио:
makschocomint.ru

