Камалов Артур

Мужчина, 21 год, родился 17 августа 2004

+7 (996) 3693380 — предпочитаемый способ связи
camalov.arthur@gmail.com
telegram: @artur_dot_hpp

Проживает: Москва
Гражданство: Россия, есть разрешение на работу: Россия
Готов к переезду, не готов к командировкам

Желаемая должность и зарплата

React / Next.js frontend developer

Специализации:
— Программист, разработчик
Тип занятости: полная занятость
Формат работы: на месте работодателя
Желательное время в пути до работы: не имеет значения

Опыт работы — 4 года 1 месяц

Декабрь 2023 — Октябрь 2025
1 год 11 месяцев

Composoft

Angular frontend developer

Я как React frontend developer отвечал за то, чтобы тяжелые и неповоротливые бизнес-приложения не бесили пользователей, а летали. Я не просто «крашу кнопки», а продумываю сложную инженерку под капотом, чтобы всё это добро не развалилось через месяц и легко масштабировалось. Плюс присматриваю за командой, чтобы код был чистым, а костылей — поменьше

Мои главные достижения:

1. Ускорил работу портала с огромным каталогом товаров
Проблема: В системе было более 120 000 товаров. Из-за этого огромные таблицы и списки заказов открывались по 5–6 секунд, а менеджеры постоянно натыкались на ошибки.
Что сделал: Перенастроил логику отображения данных так, чтобы программа не пыталась загрузить всё сразу, а показывала только то, что нужно в данный момент. Исправил технические ошибки, из-за которых программа «съедала» память компьютера.
Результат: Все стало работать в 3-4 раза быстрее (загрузка менее 1.5 сек). Ошибок стало в 2 раза меньше, а менеджеры по продажам стали работать эффективнее и меньше жаловаться./

2. Создал конструктор для быстрого запуска новых клиентов

Проблема: Для каждого нового заказчика приходилось переписывать программу почти с нуля, что занимало много времени.
Что сделал: Разработал универсальную систему («UI-кит»), которую можно легко подстраивать под разные требования, просто переключая «рычажки» (настройки), не меняя основной код.
Результат: Теперь запуск нового клиента занимает на 40% меньше времени.

3. Навел порядок в «мозгах» приложения (управлении данными)
Проблема: Данные внутри программы передавались хаотично, программисты путались в коде, из-за чего исправление одной детали ломало другую.
Что сделал: Внедрил четкий и понятный state manager (Redux) с понятной документацией.

Использовать клиентские данные стало гораздо проще
Задачи стали закрываться практически в два раза быстрее, потому что код стал понятным и предсказуемым.

4. Сделал работу с документами и заказами удобнее
Проблема: Клиенты часто ошибались при заполнении сложных форм заказа или массовом редактировании товаров.
Что сделал: Переделал формы так, чтобы они моментально проверяли ошибки и подсказывали пользователю, что не так.
В следствии пользователи улучшился опыт юзеров, а жалоб в службу поддержки стало поступать меньше

5. Техническое обновление и стабильность
Что сделал: Раз в полгода обновлял технологии до последних версий, настроил линтеры, prettier, tslint и husky
Благодаря этому проводит код ревью стало проще а в прод стало попадать гораздо меньше багов

Навыки:
Использую React (современные версии 13–17) и TypeScript.
Умею строить сложные системы для больших нагрузок, которые легко масштабировать на тысячи пользователей.
Проверяю чужой код, обучаю коллег и принимаю важные технические решения по улучшению фронтенд приложений

Апрель 2022 — Сентябрь 2023
1 год 6 месяцев

Securige

React Frontend Developer

Работал над созданием софта для систем безопасности. Основной задачей было сделать так, чтобы интерфейс для операторов видеонаблюдения не «умирал» при выводе десятков камер и огромного потока событий в реальном времени.

Что конкретно делал:

Тянул React Web приложение на Electron. Это была программа для операторов. Занимался интеграцией фронта с «начинкой» системы через IPC. Настраивал так, чтобы видео крутилось плавно, а интерфейс при этом не тормозил.

Оживил видео-сетку. Пилил функционал для работы с архивами и живыми потоками. Большой проблемой были утечки памяти и нагрузка на процессор. Решал эти проблемы через оптимизацию React-компонентов и правильную работу с RxJS.

Сделал формы и настройки камер простыми в использовании. В таких системах админки обычно ужасные, я же старался превратить их в удобные инструменты с нормальной валидацией и понятной логикой прав доступа.

Обработка алертов. Реализовал систему уведомлений о событиях безопасности. Всё должно было работать мгновенно (через WebSocket), чтобы оператор не пропустил момент кражи или взлома.

С чем работал (стек):
React (17), TypeScript, Electron (IPC, main/renderer), RxJS (много асинхронщины), SCSS, REST и WebSocket.

Чего добился:

Собрал с нуля несколько критически важных модулей для мониторинга, которые сейчас реально работают на объектах.
Справился с проблемой «лагающего интерфейса» при большом количестве камер в одном окне.
Причесал код, вынес общую логику в переиспользуемые сервисы, чтобы новые фичи не ломали старые.

Август 2021 — Апрель 2022
9 месяцев

Fuse8 / Freelance

Frontend Developer

— С чистого листа верстал страницы и UI-киты по макетам из Figma с помощью html и css и jsx. Следил, чтобы всё «пиксель-в-пиксель» и при этом не разваливалось на мобилках.

— Бизнес фичи писал на react хуках (в основном useState, useEffect, useMemo), стараясь держать компоненты просытми чистыми и не перегруженными.

— «Скрещивал» фронтенд с бэкендом: прописывал запросы к REST API обрабатывал всевозможные ошибки сервера.

— Для логики оформления заказов создавали реактивные формы с валидацией данных и сохранением черновика

— Постоянно рефакторил старый код. Если видел спагети код — причесывал и оптимизировал чтобы всё летало и проект оставался легко масштабируемым.

— Задачи обсуждали на созвонах с заказчиками и дизайнерами. Не просто «красил кнопки» а обсуждал как сделать интерфейс удобнее для людей.

Образование

Высшее

2022
Высшее

Представительство Южно-Уральского государственного университета (Национального исследовательского университета), Южноуральск

Математики, механики и компьютерных наук, Разработчик программного обеспечения

Навыки

Знание языков

Русский — Родной

Навыки

React TypeScript JavaScript Redux CSS3 HTML5 REST API Git Next.js Node.js Docker UI / UX Webpack PWA Websockets Clean Architecture ООП SOLID KISS DRY SCSS Tailwindcss GraphQL Zustand React-Query Pixel-perfect GitLab CI / CD Jenkins Jest React Testing Library

Дополнительная информация

Обо мне

Портфолио: https://react-portfolio-roan-chi.vercel.app/

Обо мне:

React-разработчик, опыт - 4+ лет. В основном занимался сложными Enterprise и B2B штуками. Люблю делать SPA, которые не «рассыпаются» при масштабировании и которые потом не больно поддерживать.

За плечами опыт с высоконагруженными системами: каталоги на 100тыс. + позиций, многофункциональные таблицы и формы. Всегда стараюсь смотреть не только в код, но и на бизнес — чтобы UX не страдал, страницы грузились быстро, а техдолг не копился мертвым грузом.

Стек:

React, TypeScript, Next.js.
State-management: Redux или Facade-подход (выбираю по задаче).
Делал как веб-сервисы, так и мобилки на React Native и десктоп на Electron.
Не просто пишу код, а участвую в архитектуре, провожу ревью и подтягиваю джунов.
Люблю, когда код предсказуемый и чистый, а результат можно пощупать в цифрах. Ищу команду, с которой можно вдолгую развивать крутой продукт.

Связь и исходники:

Портфолио: https://react-portfolio-roan-chi.vercel.app/
GitHub: https://github.com/artur-kamalov?tab=repositories
Telegram: @artur_dot_hpp

Технологии и инструменты React:

Языки и основы: JavaScript (ES6+), TypeScript, JSX/TSX, HTML5, CSS3, SCSS/SASS
Сборка и окружение: Vite, Webpack, Babel, npm, yarn, pnpm
Управление состоянием: React Hooks, Context API, Redux Toolkit, Zustand
Работа с данными: REST API, Axios, Fetch, React Query, GraphQL, Apollo Client
Маршрутизация: React Router, Next.js App Router
Стилизация и UI: Tailwind CSS, Material UI (MUI), Styled-components, Emotion, Chakra UI
Формы и валидация: React Hook Form, Formik, Yup, Zod
Тестирование: Jest, React Testing Library, Cypress, Playwright
Дополнительно: Storybook, Framer Motion, i18next, Lodash, date-fns/dayjs, Socket.io