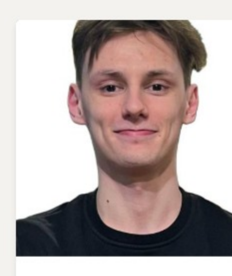

Олег Литвинчук

Fullstack-разработчик

Профиль

Fullstack-разработчик с опытом в высоконагруженных сценариях, где я отточил навыки мгновенного анализа и принятия решений под давлением. Владею современными веб-технологиями (HTML/CSS/JS, React, Node.js), применяя их в личных проектах для создания динамичных приложений и интеграций. Готов внести дисциплину, стратегическое мышление и надежность в команду разработчиков на позиции frontend/fullstack-разработчика. Стремлюсь к инновационным вызовам и быстрому росту в IT, превращая сложные задачи в эффективные решения.

Ссылки на мои проекты:
npmjs: https://www.npmjs.com/~sgraph
GitHub: https://github.com/Shadowgraph-1/
AI Assistant System: https://ai-forall.netlify.app/
Проекты на заказ:
Ногтевой сервис: https://lillnails.netlify.app/

Образование

Fullstack-разработчик — янв. 2024 - янв. 2026
Нетология

Оператор диспетчерской (производственно-диспетчерской) службы — сент. 2023 - сент. 2024
МосТех

Информационные системы и технологии (Технологии искусственного интеллекта) — сент. 2024 - сент. 2026
МосТех

Комплексная защита объектов информатизации — сент. 2023 - авг. 2027
ИПО

Опыт работы

Fullstack-разработчик
Опыт работы 4.6 лет

- ***Владение современными веб-технологиями:*** HTML5, CSS3, JavaScript, Python, React, Node.js, TypeScript
- ***Разработка и поддержка нескольких проектов:*** персональный сайт-портфолио, AI Assistant System, сайт для салона ногтевого сервиса Lillnails, административная панель Nails Days Admin Panel
- ***Интеграция с внешними API и сервисами:*** NPM Registry API, Firebase, Telegram API, Netlify API
- ***Использование современных инструментов и библиотек***: Vite, Tailwind CSS, Framer Motion, React Router, ESLint, Axios, React Hook Form, Material-UI, React DatePicker, DayJS, PapaParse, JSZip, Tauri, Rust
- Разработка SPA приложений с адаптивным дизайном и анимациями
- Оптимизация производительности, обеспечение безопасности и удобства использования
- Реализация ролевой модели и аутентификации через Firebase Auth
- Деплой и CI/CD с использованием Netlify
- Создание кросс-платформенных десктопных приложений с Tauri

Личные данные

Дата рождения
25 февраля 1998 г.

Веб-сайт
litvinchuk-resume.netlify.app

Рабочий стек

HTML5
CSS3
JavaScript
React
Python
React DatePicker
Firebase Auth
Google Firebase
Tauri
PapaParse
JSZip
Telegram API
Netlify API
NPM Registry API
Tailwind CSS
Material-UI
Lucide React
Git
npm
RESTful APIs (fetch/Axios)
TypeScript
ESLint
React Router
React Hook Form
PostCSS/Autoprefixer
React Firebase Hooks
Node.js
Prettier
Vitest
Framer Motion
React Router DOM
Zustand
Axios

Языки

Английский

Курсы

HTML и CSS: основы веб-верстки
JavaScript: основы работы в браузере
JavaScript: углубленный курс веб-разработки
Расширенный инструментарий JavaScript в браузере
React: Библиотека №1 в современной фронтенд-разработке
Основы языка программирования Python
Git — система контроля версий
ООП и работа с API
Базы данных для python-разработчиков
Python для проектов веб-разработки
Профессиональная работа с Python
Django: создание backend-приложений
Поколение Python: Курс для начинающих
Введение в язык Python
Основы ООП
Структура данных и процедурное программирование
ChatGPT for Everyone, OpenAI

Проекты

Персональный сайт-портфолио — янв. 2025 - сент. 2025
Одностраничное веб-приложение (SPA) с адаптивным дизайном
Ключевой функционал:
- Навигация: Фиксированный левый сайдбар с иконками социальных сетей и меню, плавная прокрутка к секциям с активным выделением.
- Hero секция: Приветствие, фото, цитата, кнопки скачивания CV и связи, динамическая загрузка данных о пакетах из NPM API.
- About секция: Биография, цитата Илона Маска, soft skills с hover-эффектами, timeline с проектами и курсами (интерактивный с анимациями).
- Projects секция: Карточки проектов с описаниями, изображениями (GIF/PNG), ссылками на репозитории и демо.
- Portfolio секция: Карточки услуг (React Developer, Python Developer) с hover-эффектами и описаниями.
- Skills секция: Список технологий с иконками и уровнями (Advanced/Proficient).
- Contact секция: Форма связи с валидацией, социальные ссылки, анимации при появлении.
- Адаптивность: Responsive дизайн для мобильных/планшетов/десктопов, скрытие элементов на малых экранах.
- Интеграции: Загрузка данных о NPM-пакетах (версия, скачивания, дата обновления)
- Анимации: Плавные переходы с Framer Motion (подъем меню, появление секций), CSS transitions для hover.

Используемые технологии и стек Frontend:
- React (компоненты, хуки, управление состоянием)
- Vite (сборка и dev-сервер)
- Tailwind CSS (стилизация, утилитарные классы, responsive дизайн)
- Framer Motion (анимации: motion.div, spring transitions)
- Lucide React (иконки: Download, Mail, Code, Zap и др.)

Интеграции и API:
- NPM Registry API (загрузка метаданных пакетов: версия, скачивания, дата)
- Telegram/VK/GitHub API (ссылки на профили)

Дополнительные инструменты:
- ESLint (линтинг кода, конфигурация в eslint.config.js)
- React Router (для навигации, хотя сайт SPA без маршрутизации)
- date-fns (работа с датами в timeline)
- React Hook Form (для форм в Contact, если используется)
- Axios (для HTTP-запросов к API)

Другие зависимости:
- React DOM, React Strict Mode, PostCSS/Autoprefixer (из конфига).

Архитектура и особенности реализации:
- **Модульная структура:** Код разделен на компоненты (Header, Hero, About и т.д.) в папке components/, хуки в hooks/ (useSectionAnimation для анимаций секций).
- **Состояние и данные:** useState/useEffect для загрузки данных (NPM API), локальное состояние для активных секций и форм.
- **Стилизация:** Утилитарный подход с Tailwind (классы в JSX), глобальные стили в index.css (padding-left для левого сайдбара, анимации shine).
- **Адаптивность:** Медиа-запросы в Tailwind (md:, lg:), flexbox/grid для layout, min-width в body для предотвращения сжатия.
- **Оптимизация:** Ленивая загрузка компонентов, оптимизация изображений (object-cover), debounce для скролла.
- **Безопасность и UX:** Валидация форм, обработка ошибок API, плавные анимации без перегрузки.
- **Деплой:** Vite build для production, поддержка SPA (index.html как entry point).
- **Вклад и достижения**
- **Разработал сайт с нуля:** от wireframe до рабочего продукта, интегрировав API и анимации для привлекательного UX.
- **Реализовал интерактивные элементы:** timeline с курсами, динамическую загрузку данных о проектах, адаптивный хедер с меню.
- **Обеспечил производительность**: оптимизированные запросы к API, плавные анимации без лагов, быстрая загрузка (Vite).
- **Продемонстрировал навыки:** React экосистема, интеграция API, CSS анимации, управление состоянием — сайт служит живым примером fullstack-возможностей.

React CLI Tool — май 2023 - Сейчас
Командно-строчный инструмент для создания и настройки React проектов с поддержкой нескольких фреймворков и Tailwind CSS v4
Основные технологии
- **Node.js** - среда выполнения
- **JavaScript (ES Modules)** - основной язык разработки
- **Inquirer.js** - интерактивные командные промпты
- **Ora** - терминал спиннеры для индикации прогресса
- **Chalk** - цветной вывод в терминал
- **Fs-extra** - расширенная работа с файловой системой
- **Child_process** - выполнение системных команд

Поддерживаемые фреймворки и библиотеки
- **React 19** - основной фреймворк
- **Vite** - сборщик и dev-сервер
- **Next.js** - full-stack React фреймворк
- **Remix** - современный full-stack фреймворк
- **Gatsby** - статический сайт генератор
- **Tailwind CSS v4** - утилитарный CSS фреймворк

Инструменты разработки
- **ESLint** - линтинг кода
- **Prettier** - форматирование кода
- **Husky** - Git хуки
- **Lint-staged** - запуск линтеров только на staged файлах
- **Vitest** - тестовый фреймворк
- **@testing-library/react** - утилиты для тестирования React компонентов

Дополнительные библиотеки
- **Lucide React** - иконки
- **Framer Motion** - анимации
- **React Router DOM** - клиентский роутинг
- **Zustand** - управление состоянием
- **React Hook Form** - формы
- **Axios** - HTTP клиент

Реализация
Архитектура проекта
Проект разделен на модули для лучшей поддерживаемости:
- bin/cli.js - точка входа
- lib/ - основные модули (main.js, customizers.js, installers.js, prompts.js, logger.js, i18n.js)
- steps/ - шаги установки (language-selection.js, project-setup.js, installation.js, customization.js, directory-handling.js)

Ключевые возможности:
Многоязычный интерфейс:
- Поддержка английского и русского языков
- Автоматическое определение языка интерфейса
- Локализованные сообщения и промпты

Пресеты установки:
- **Minimal**: только React + Vite + Tailwind CSS
- **Full**: полный набор (роутинг, состояние, формы, UI, тестирование, инструменты)
- **Testing**: инструменты для тестирования
- **Custom**: ручной выбор библиотек и инструментов

Автоматическая настройка:
- Создание структуры папок проекта
- Настройка .env файлов для разных фреймворков
- Генерация .gitignore
- Установка и конфигурация ESLint, Prettier, Husky
- Настройка Git хуков для автоматического линтинга
- Удаление встроенных ненужных компонентов

Интерактивные промпты:
- Предупреждения о долгой установке для Full пресета
- Обработка ошибок и повторные попытки

Обработка ошибок:
- Логирование ошибок в файл setup-log.txt
- Graceful degradation при неудачных установках
- Информативные сообщения об ошибках

Особенности реализации:
Модульная архитектура
Каждый шаг установки вынесен в отдельный модуль, что позволяет легко добавлять новые возможности и поддерживать код.

Асинхронная обработка
Использование async/await для корректной обработки асинхронных операций установки пакетов и выполнения команд.
Кросс-платформенность
Инструмент работает на Windows, macOS и Linux благодаря использованию Node.js и кросс-платформенных команд.
Безопасность
- Валидация вводимых данных
- Безопасное выполнение системных команд
- Правильная обработка путей к файлам

AI Assistant System — июнь 2024 - янв. 2025
Веб-приложение (SPA) для демонстрации возможностей AI-ассистента разработчика
Технологии и стек:
- React (функциональные компоненты, хуки)
- Vite (быстрая сборка и dev-сервер)
- Tailwind CSS (адаптивная стилизация, эффекты)
- CSS-модули и кастомные стили (effects.css)
- SVG/JSON-ассеты для графики и шрифтов
- Простая архитектура с разделением на компоненты и страницы

Реализовано:
- **Минималистичный дизайн** в стиле современного IDE (VS Code), с акцентом на тёмную тему и cyan-цвета
- **Hero-секция** с анимированным звёздным фоном, интерактивной матрицей и навигацией
- **Outcome Showcase** — витрина преимуществ продукта с анимированными метриками, "до/после" и шкалой успешности по языкам
- **Список поддерживаемых языков** с официальными SVG-иконками (Python, JS, TS, Java, Go, C#, C++, PHP, Ruby, Rust, Kotlin, Swift, Dart)
- **Секция "КОМАНДЫ"** — демонстрация ключевых AI-команд (установка, исправление, объяснение, оптимизация, ревью, тесты)
- **Интерактивные подсказки и тултипы** для сценариев и команд
- **Многостраничная структура** (HelloWorld, RoadMap, ProjectNews, SubscriptionPlans, AIBiography)
- **Адаптивная верстка** для десктопа и мобильных устройств
- **Использование кастомных шрифтов** (helvetiker_regular.typeface.json)
- **SVG-ассеты** для оформления (Neon_Kami.png, Shadow_Oni.png)

Особенности:
Нет серверной части — только фронтенд
Чистый, структурированный код, легко расширяемый
Используются современные best practices React и CSS

Сайт для салона ногтевого сервиса Lillnails — март 2021 - март 2024
Одностраничное веб-приложение (SPA) для клиентской стороны салона, frontend разработчик с интеграцией backend-сервисов
Ключевой функционал
Главная страница:
Градиентный фон с анимированными SVG-элементами, фиксированный header с навигацией, секции "Услуги", "Галерея", "Отзывы", "Правила записи".
Услуги:
Переключатель категорий (Наращивание, Коррекция, Маникюр) с динамическим отображением размеров/типов и цен из статических данных.
Галерея:
Сетка изображений из папки gallery.
Отзывы:
Имитация чата с реальными сообщениями и декоративными элементами (сердечки).
Правила записи:
Информационные карточки о предоплате, отмене, возрасте и т.д.
Форма записи:
Модальное окно с полями для имени, телефона (автоформатирование), Telegram, услуги, размера, даты/времени. Сохранение в Firebase, обновление слотов, поддержка промокодов (скидки из Firestore), отправка уведомлений в Telegram.
Подтверждение записи:
Модальное окно с чеком и инструкциями по предоплате.
Maintenance mode:
Проверка статуса из Firestore, dev-режим по паролю (localStorage), блокировка сайта для пользователей.
Адаптивность:
Медиа-запросы для мобильных и планшетов, скрытие меню на малых экранах.

Используемые технологии и стек
Frontend:
- React (компоненты, хуки, управление состоянием)
- Vite (сборка и dev-сервер с плагином React)
- Tailwind CSS (стилизация, responsive дизайн)
- AOS (Animate On Scroll) для плавных анимаций при прокрутке

База данных и интеграции:
- Firebase Firestore (хранение записей, дат, промокодов; конфигурация в firebase.js)
- Telegram Bot API (уведомления о новых записях; токен и чат-ID закодированы)

Дополнительные инструменты:
- ESLint (линтинг кода)
- date-fns (работа с датами)
- JSZip (архивация файлов)
- Netlify (развертывание)

Архитектура и особенности
- **Модульная структура:**

Код организован вокруг секций главной страницы, с использованием модальных окон для форм.
- **Интеграция данных:**

Загрузка доступных дат и промокодов из Firebase при монтировании компонентов, обработка ошибок и сброс форм после действий.
- **Безопасность и UX:**

Защита dev-режима паролем, автоформатирование полей, имитация отзывов для вовлеченности.
- **Деплой:**

Автоматическое развертывание на Netlify с поддержкой SPA (redirects).
- **Адаптивность:**

CSS-медиа-запросы для кросс-устройственной совместимости.

Вклад и достижения:
- Разработал полное SPA от концепции до запуска, интегрировав Firebase и Telegram для автоматизации.
- Реализовал интерактивные элементы и формы с валидацией, обеспечивая удобство записи и уведомлений.
- Обеспечил адаптивность и производительность, оптимизировав загрузку данных и анимации.
- Проект демонстрирует навыки в React, интеграции с облачными сервисами и создании пользовательских интерфейсов для бизнеса.

Административная панель для салона ногтей (Nails Days Admin Panel) — апр. 2022 - май 2023
Веб-приложение с десктопной версией, (Frontend + Backend интеграция + Десктоп адаптация)
Ключевой функционал:
- **Управление записями:** Просмотр, фильтрация, изменение статуса (ожидает оплаты, предоплата, оплачен и т.д.), перенос записей на другие даты/время.
- **Управление датами и временем:** Добавление/редактирование доступных слотов, отображение занятости.
- **Промокоды:** Генерация, управление скидками, отслеживание использования.
- **Аналитика:** Расчет доходов (дневной, недельный, месячный, общий), топ услуг, экспорт данных в CSV, очистка истории.
- **Техническое обслуживание:** Включение режима "техработ" с уведомлениями в Telegram, деплой файлов на Netlify, управление паролями для dev-режима.
- **Аутентификация:** Вход через Firebase Auth с ролевой моделью (администратор/мастер).
- **Реальное время:** Синхронизация данных через Firestore с мгновенными обновлениями интерфейса.

Используемые технологии и стек Frontend:
- React
- Vite
- Tailwind CSS (стилизация, responsive дизайн)
- React DatePicker (выбор дат с русской локализацией)
- DayJS (работа с датами и временем)
- Material-UI (дополнительные компоненты, если используются)

Backend и база данных:
Firebase (Firestore для хранения данных, Firebase Auth для аутентификации, Analytics для метрик)
Десктопная версия:
- Tauri (фреймворк для создания кросс-платформенных десктопных приложений на базе Rust и веб-технологий)
- Rust (backend для Tauri, конфигурация в Cargo.toml)

Дополнительные инструменты:
- JSZip (архивация файлов для деплоя)
- PapaParse (экспорт данных в CSV)
- Telegram API (уведомления о техработах)
- Netlify API (деплой и управление сайтом)
- ESLint (линтинг кода)
- PostCSS/Autoprefixer (обработка CSS)

Другие зависимости:
- React Firebase Hooks
- React Strict Mode
- ThemeProvider для Material-UI.

Архитектура и особенности
- **Модульная структура:** Код разделен на страницы (pages/), каждая отвечает за отдельную панель (Dashboard, BookingsPanel и т.д.).
- **Реактивность:** Использование onSnapshot для реального времени обновлений из Firestore.
- **Безопасность:** Защищенные маршруты (ProtectedRoute), ролевая аутентификация, хранение чувствительных данных в .env.
- **Деплой и CI/CD:** Интеграция с Netlify для автоматического деплоя, поддержка _redirects для SPA.
- **Кросс-платформенность:** Tauri позволяет собирать приложение для Windows, macOS и Linux.
- **UI/UX:** Адаптивный дизайн с помощью Tailwind, всплывающие подсказки (HelpIcon), модальные окна для действий.

Вклад и достижения
- Разработал полный цикл от идеи до рабочего приложения, включая интеграцию с Firebase и Tauri.
- Реализовал сложную логику управления записями и аналитикой, обеспечивая точность расчетов и удобство использования.
- Оптимизировал производительность с помощью реального времени обновлений и эффективной работы с данными.
- Обеспечил масштабируемость и безопасность, добавив ролевую модель и защиту от несанкционированного доступа.