

PORTFOLIO



CONTACTS & LINKS

TELEGRAM

@hoursen

EMAIL

cherepov.lab@gmail.com

LINKEDIN

linkedin.com/in/nikita-cherepov

САЙТ

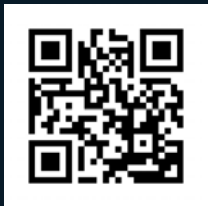
ncherepov.ru

GITHUB

github.com/NikitaCherepov

LOCATION

Tomsk, Russia



SCAN TO VIEW PORTFOLIO

PRIMARY STACK

React / Next.js
TypeScript / JavaScript
SCSS / CSS
TanStack Query
MobX / Zustand
Framer Motion
Payload CMS / Strapi
Node.js / REST API
PostgreSQL / SQLite

NIKITA CHEREPOV

FRONTEND / FULLSTACK WEB DEVELOPER

ABOUT:

Frontend / Fullstack developer with 2+ years of commercial experience. My core stack includes React, Next.js, and TypeScript. I build complex interfaces, interactive web applications, e-commerce websites, landing pages, animations, and CMS integrations. I also develop my own fullstack projects.

COMMERCIAL EXPERIENCE

CreaGeese IT

April 2024 - July 2026

2 years 3 months

Key responsibilities:

- Developed and maintained commercial projects, including e-commerce websites, complex landing pages, and custom solutions based on Next.js and React. Integrated Payload CMS and Strapi.
- Designed interactive tables and filters. Managed application state with MobX and Context and configured request caching with TanStack Query.
- Created complex animations and interactive elements using Framer Motion.
- Refactored code, created libraries of reusable UI components, and ensured responsive cross-browser compatibility.
- Collaborated with designers and backend developers in a cross-functional team and maintained a branch-based development workflow.

PROJECTS I CONTRIBUTED TO:

Sauna Design Website Builder

3-dims.3-dims.ru/ru



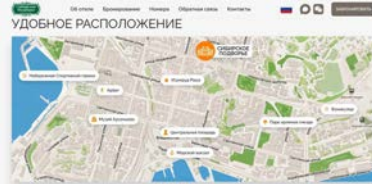
Equipment Store

rspa-pro.ru



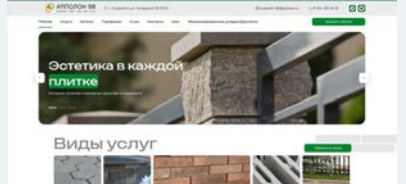
Hotel Landing Page

otelsp.com



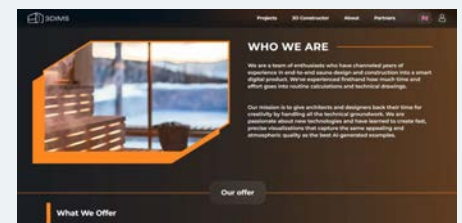
Tile Store

www.apollon98.ru



Landing

3-dims.eu



Metalamp

February 2022 - May 2022

3 months

Key responsibilities:

- Developed an MVP using Bubble no-code platform.
- Designed user roles and the database structure.

Website: responsiveagent.bubbleapps.io/version-test/

OWN PROJECTS

1. CHATTER

AI AGENT SYSTEM

A multiplatform AI agent system that can be deployed with a single Docker command.

Stack: TypeScript, React, Next.js, Electron, Node.js, Vite, TanStack Query, SQLite, REST API, WebSockets, Docker, Docker Compose, GitHub Actions, Telegram Bot API / Telegraf, Whisper.cpp, Piper TTS, FFmpeg, i18next, Framer Motion.

Architecture and infrastructure

- Designed and deployed a system consisting of six services: Backend API, Electron client, Telegram bot, Webapp, Voice Service, and Admin Panel, orchestrated with Docker Compose.
- Implemented session synchronization for seamless real-time access to chats, agent state, and PC controls across Telegram and the desktop application using WebSocket/SSE.

LLM engine and optimization

- Implemented model routing through OpenRouter, DeepSeek, Xiaomi, and other providers, including automatic token cost calculation, fallback providers in case of API failures, and dynamic context archiving.
- Reduced API costs by achieving a 90-95% cache hit rate for input tokens through stabilized system prompts.
- Implemented task delegation between models: the primary LLM automatically invokes Vision models for image analysis and lightweight models for command risk assessment and chat naming.

Desktop and system integrations

- Built operating system integrations, including CLI command execution, Cyrillic encoding through Base64, file-system reading and editing, screenshot capture, and webcam access.
- Added local TTS using Piper and Chrome Native alongside cloud providers such as Cartesia, as well as wake-word detection powered by Whisper.

Security and administration

- Designed a secure execution architecture. The AI can manage servers over SSH and interact with local computers, while every action requires user confirmation through the interface.
- Protected sensitive data: SSH keys, email credentials, and passwords are encrypted and never passed into the agent context. Added prompt-injection protection and AI-assisted review of potentially dangerous commands.
- Developed an admin panel for managing roles, subscription budget limits, logs, and system load.
- Automated localization with an i18n synchronization script that translates all system components through LLM calls.

Agent capabilities

Vector memory

Voice and wake-word

Email management

Smart home

Web search and page parsing

Images generation

Subagents and spec-agents

Telegram integration

SSH management

Macros and runbooks

Custom prompts and AI editors

Reminders and Cron jobs

Pixel avatar

Built-in maps and route search

Remote PC control

Telegram voice-message transcription

Embedded browser

Notes

Message regeneration and branching

Document processing and video compression

GITHUB

github.com/NikitaCherepov/chatter

2. Music List - SaaS for release planning

A service for musicians and labels featuring a catalog of tracks, albums, and artists, a release calendar, analytics, subscriptions, and an admin panel. It stores metadata and schedules and streamlines release management.

- Designed data models and relationships for users, tracks, albums, artists, subscriptions, notifications, gifts, and logs.
- Implemented virtualized track and album tables using @tanstack/react-virtual, including filtering, search, and inline editing.
- Built a drag-and-drop release calendar. Moving releases or albums automatically shifts the associated tracks.
- Developed analytics with period-based aggregation and filters. The API generates empty intervals to ensure accurate chart rendering.
- Implemented subscription plans and limits for FREE, PLUS, and PRO tiers, including a PRO trial and upgrades or downgrades without losing the paid period.
- Integrated payments through Selfwork and webhooks, including admin-side order tracking and refunds.
- Added a multilingual interface and AI-powered notification localization with translation caching.
- Added a Redis layer for bans, email rate limiting, and session caching to reduce database load.
- Optimized large operations with chunked uploads and imports and protection against bulk updates affecting tens of thousands of records.

Stack: Next.js, React, TypeScript, Prisma, PostgreSQL, Redis, next-intl, dnd-kit, @tanstack/react-virtual, Chart.js.

Website: music-list.ru/en/about

OWN PROJECTS

3. Portfolio website with an admin panel

My working tool for showcasing development and music projects.

- Designed a flexible page and component architecture that supports rapid project growth.
- Implemented PageTransition for animated slide-style navigation from the menu.
- Built a compact draggable audio player with a dedicated mobile mode.
- Polished responsive behavior for desktop and mobile devices.
- Developed an admin panel for adding and editing projects. Added a guest preview mode with API-level access restrictions.
- Created a complex reusable card component with smooth animated transitions between table and square-grid layouts using Framer Motion.

Stack: Next.js, React, TypeScript, Framer Motion, SCSS, dnd-kit, Prisma/PostgreSQL, JWT-auth.

GitHub: github.com/NikitaCherepov/my-portfolio

Website: ncherepov.ru

4. ShareThisLoot - website for a D&D loot distribution

A website that automates currency conversion and loot distribution in D&D.

- Multilingual support and local data storage
- Responsive interface
- Greedy currency distribution algorithm

Stack: Next.js, React, TypeScript, MobX, next-intl

GitHub: github.com/NikitaCherepov/share-this-loot

Website: sharethisloot.ru/ru

TECHNICAL PROFILE

FRONTEND

- ReactNext.jsViteTypeScript
- JavaScriptSCSSFramer Motion

STATE / DATA

- TanStack QueryMobXZustand

BACKEND

- Node.jsREST APIWebSockets
- PostgreSQLPrismaSQLiteRedis
- SSE

CMS

- Payload CMSStrapi

INFRASTRUCTURE

- DockerGitLinux

ADDITIONAL

- ElectronThree.jsdnd-kiti18n
- TelegrafChart.js