Roman Ermishin

Software Developer

Контакты

- **GitHub:** @EternalRival
- **E-Mail:** erdevelopment8@gmail.com
- **Discord:** eternal_rival
- **Telegram:** @Eternal_Rival

Обо мне

Увлёкся программированием в июле 2020 года в качестве хобби. Самостоятельно изучил некоторые основы, разработал приложение IQTestRPM (Raven's Progressive matrices) и выпустил его в Google Play (1,5k установок и 4.54★). Когда количество загрузок стало расти, я понял, что мне нравится создавать приложения, которыми пользуются люди. На некоторое время забросил это хобби по личным причинам.
В 2022 году вернулся к разработке, начал самостоятельно изучать JS, а затем и остальные технологии Frontend стека (и немного Backend).
Прошел курсы JS, Node и React в RSSchool (курсы от EPAM), участвовал в командных проектах и являюсь волонтёром (помогаю кураторам с организацией заданий и новичкам с различными вопросами).

Технологический стек

- Git
- JavaScript / TypeScript / React
- Webpack / Vite / NextJS (Pages Router)
- TanstackRouter / React Router
- Zustand / Redux Toolkit
- TanstackQuery / Redux Toolkit Query
- RESTful Api / GraphQL
- CSS modules / SCSS / TailwindCSS / MUI
- Yup / Zod
- ESLint / Stylelint / Husky
- Jest / Vitest / Testing Library

Проекты

- **decision-making-tool**
  - Пет-проект
  - source / deploy
  - Приложение для стримеров, контентмейкеров и любителей настольных игр.

    - Потребовался инструмент для случайного выбора действия (выбор фильма для просмотра или игры) и/или розыгрыша призов.
    - Нужно было приложение для взвешенной жеребьёвки.
    - Написал двухстраничное приложение с роутингом. Реализовал интерфейс для добавления/редактирования названий лотов и их "весов". Реализовал сохранение введённых данных между сессиями с помощью `Storage Api`. Реализовал визуальное отображение списка лотов в виде "колеса фортуны" с помощью `Canvas Api` и анимировал его вращение на случайную позицию, используя `requestAnimationFrame`. Реализовал сохранение списка лотов в файл и загрузку из файла.
    - Приложение используется для случайного выбора действий. Получена позитивная обратная связь.
  - TS, React, React Router, Redux + RTK, Vite, CSS Modules, Zod
- **rsschool-tools**
  - Пет-проект
  - source / deploy
  - Приложение для студентов и активистов RSSchool.
    - Зачастую студентам требуется помощь по исправлению ошибок в их приложениях или поиску деплоя их приложения на сервисе `github.io`.
    - Нужно было приложение для быстрого поиска ссылок на деплои студентов.
    - Я разработал приложение, которое ожидает на ввод необходимые для поиска данные, составляет ссылки из наиболее популярных вариантов ссылок и ищет среди них рабочую ссылку на деплой. Реализовал дополнительные фичи, среди которых "предварительный подсчет баллов за Crosscheck-проверки".
    - Благодаря приложению стало возможным быстро найти рабочую ссылку на деплой студента, осмотреть работу и оказать ему необходимую помощь, минуя дополнительные вопросы по типу "пришли ссылку на деплой", "покажи структуру папок в ветке `gh-pages`" и т.п. Получена позитивная обратная связь.
  - TS, React, NextJS (Pages Router), TailwindCSS, SWR, Zod
- **rsschool-gratitude-summary**
  - Пет-проект
  - source / demo
  - Приложение для студентов и активистов RSSchool.
    - Отсутствовала возможность получить сводку по отправленным и полученным благодарностям.
    - Нужно было приложение, которое запрашивает и парсит данные по студентам с backend.
    - Я разработал приложение для получения и обработки данных на клиенте, реализовал интерфейс в виде таблицы с фильтрацией и сортировкой. Приложение выполнено в виде расширения для Chrome и предназначено для запуска на платформе app.rs.school.
    - Благодаря приложению стало проще выявлять наиболее активных студентов для их выдвижения на роль "активист".
  - TS, Vue3, Vite, TailwindCSS, Tanstack Query, Zod, Chrome Api
- **christmas-shop**
  - source / deploy / figma

  - Двухстраничный лендинг, созданный на основе макета из Figma. Адаптивный дизайн. PixelPerfect. Компонентный подход.
  - TS, Webpack, CSS Modules
- **virtual-keyboard**
  - source / deploy
  - Приложение для набора текста с помощью физической и/или виртуальной клавиатуры.
  - JS, Webpack, Sass
- **home-library-service**
  - source
  - Backend приложение для хранения данных об артистах, альбомах и песнях с возможностью редактирования списка "избранного". Реализован запуск приложения и базы данных в Docker-контейнерах и взаимодействие между ними. Реализована авторизация с использованием JWT. Реализованы маршруты для CRUD-операций через REST Api.
  - TS, NestJS, Docker, Docker Compose, TypeORM, PostgreSQL, JWT, Swagger
- **e-commerce-app**
  - source / deploy
  - Frontend приложение онлайн-магазина с использованием CommerceTools service (GraphQL Api) в качестве backend. Изначально это учебный командный проект RSSchool, но я выполнял его один, вне курса (в свободное время) ради расширения моего технологического стека. По завершению trial-доступа разработка и поддержка была остановлена.
  - TS, React, NextJS (Pages Router), Vitest, TailwindCSS, Material UI, Zustand, Tanstack Query, Zod, Axios, GraphQL, React Hook Form, Swiper, Feature Sliced Design, CommerceTools

Сертификаты

- RSSchool JSFE2022Q3 (Certificate)
- RSSchool NodeJS2023Q2 (Certificate)
- RSSchool React2023Q4 (Certificate)