



Feigin Lia-Aleksandra

Lead Full-Stack Engineer

IT generalist experienced in software architecture and a wide range of web-related technologies.

TypeScript | React | NestJS

f@lilit.io | 26 years old

/lialefei | Moscow, Russia

Work experience
- 8 years 9 months

Proscorn

August 2021 — Present
(1 year 4 months)
Technology BP

January 2021 — July 2021
(7 months)
Lead Engineer

July 2020 — December 2020
(6 months)
Senior Front-End Engineer

web application development agency

I took part in the development from scratch and launching of more than 10 web applications of varying complexity, first as a front-end developer and later as a lead full-stack developer in multiple teams of up to 7 engineers.

All of the projects were written in **TypeScript**, most of them used **React** (with either **MobX** or pure **React Context** for state management and **Apollo Client**) on the front-end and **NestJS** (**TypeORM**, **PostgreSQL**, **Redis**) on the back-end with **GraphQL** (**Apollo Server**) between them.

I designed integrations with various external services (**Stripe**, **AirTable**), complex legacy systems (**Moodle**, custom enterprise solutions), built **web3**-applications with wallet connection and real-time data aggregation from multiple external sources (**BitQuery**, **CoinMarketCap**, **Covalent**), worked with **Strapi** (Headless CMS), used **React Native** to create web application wrappers so that they could be published in the stores.

As a BP, I managed company-wide transition to microservices and **Apollo Federation** (multiple independent GraphQL APIs combined into a supergraph), per-project monorepos (using **Yarn Workspaces**), introduced standards for service architecture and company-wide code re-use, streamlined application deployment process with **Docker**.

Projects

OKO

October 2019 — July 2020

OKO is an **Electron** application made for gathering and visualising data from **CSV** files and **PostgreSQL** instances located on a local network. It used **React** internally with hand-crafted **SVGs** for visualisations.

Comm

March 2018 — July 2018

Comm is a **WebRTC**-based **React** application for video and audio calls with direct messaging and file sharing via an encrypted P2P-connection using **RTCDatChannel**. A custom **STUN/TURN** server was built for it to work anywhere regardless of network conditions.

Time Task Manager

September 2017 — August 2019

Time is an offline-first **PWA** for managing tasks in a cycle. It used **React** and **Redux** on the front-end and **Node.js** on the back-end, communicating via **WebSocket** for real-time data synchronisation.

Higher School of Branding

master`s in marketing program

July 2020 — Present
(2 years 5 months)
Partner

January 2018 — June 2020
(2 years 6 months)
Technical Director

March 2014 — December 2017
(3 years 10 months)
Technical Specialist

I was initially hired as a distance education engineer responsible for operating audio and video equipment and providing live video streams using an existing platform, then started working on improving both the video production and the platform itself and by this point, the job centered on full-stack web development with the main responsibility of creating a custom Learning Management System (LMS) with live streaming capabilities, while also hiring and managing a small team of distance education engineers that took on my previous tasks.

The resulting web application was built using **React** (with **Next.js** for the public parts of the system) and **Node.js** (**Express**) with a **REST API** between them. It used a custom self-hosted media streaming server (based on **nginx-rtmp-module**) that accepted an RTMP source from OBS and transcoded it in real-time into multiple HLS targets using **ffmpeg** to create the best viewing experience for all users regardless of their browser of choice, location and connection quality. The recordings were initially saved in an **S3-like** storage later replaced by a **Vimeo** integration.

Parallel to the LMS creation, I also worked on the company's internal tooling – e.g., a **WebRTC**-based web application for using a smartphone as a wireless camera (an OBS video source).