

Roman Gurovich

(949) 338-8497 SF Bay Area + remote

romangurovich@gmail.com

<https://github.com/romangurovich>

Experienced generalist dev looking to implement and launch novel products that leverage ML. Ideal role will encourage me to contribute broadly: prepare data, train/deploy models, build APIs, develop UI components, design data-driven systems, set up pipelines, containerize infrastructure, and maintain metrics.

PROFESSIONAL EXPERIENCE

Fulcrum

Senior Software Engineer

remote

Oct 2022–April 2023

- Launched real-time chat and inbox functionality (C#, Mongo, Angular, GraphQL, Websockets)
- Built feature to allow partially inventory of manufacturing outputs (C#, Mongo, Typescript)
- Developed notifications that appear when schedule depends on outdated equipment/material

Trax Retail

Senior Full Stack Engineer

San Francisco / remote

May 2019–June 2022

- **Invoicing Service:** internal web service to handle itemized invoicing for our company's clients
 - Built & planned initial hexagonal architecture for back-end API service w/ cleanly separated domains for core logic, web, testing, infra, data (C#, Dependency Injection)
 - Trained team on how to work with event-sourced data model for invoices (EntityDB)
 - Created local dev environment for API service w/ Docker, Terraform, LocalStack, bash
 - Set up AWS infra w/ EC2, ECS, Aurora RDS, SQS/SNS, Security Groups, LB, Route53
 - Set up CI/CD w/ Github Actions, CloudFormation, Docker, AWS ECR, XUnit
 - Automated non-destructive DB migrations in CI/CD w/ EFCore CLI & PostgresDB
 - Built many of the core features on back-end service API:
 - Publishing an invoice to Stripe w/ notification to dependent systems
 - Deleting a drafted invoice w/ returning line items to billable pool
 - Presenting contract summaries w/ aggregated billable/billed items
 - Export report on invoices w/ dynamic line items per project
 - Cookie-based authorization to safeguard sensitive API routes
 - Built and planned initial architecture for front-end web app (NextJs, ReactJs)
 - Trained team on deploying front-end (Amplify, Github Actions)
 - Built several of the core pages and components on front-end web app:
 - Contract details page w/ draftable billing items
 - Buttons for publishing, drafting, deleting, voiding invoices in bulk or 1-1
 - Sortable columns for all tables listing contracts, invoices, line-items, ect.
 - Notification bubbles for succeeded/failed actions across pages
 - Filters on contract and invoice searches
- **Audit Assist:** AI-assisted merchandising mobile app feature to help contractors stock retail shelves
 - Built and planned an aggregation service to listen for events from AI pipeline, reach into the facts database, and munge the data into a digestible payload to enable sub-minute image analysis via the mobile app/API (Python, MySQL, AWS SQS/SNS, GCP PubSub)
 - Coordinated data setup across 6+ teams at the company to enable the Audit Assist integration: oversaw stores setup, tasks setup, inventory setup, user setup, AI template setup, new token-based authorization, cross-cloud resource provisioning and whitelisting

- o Launched functionality on mobile API to consume AI insights such as in-stock availability and number of facings; surfaced them to mobile app (C#, MongoDB, SQS)
- **Shipping Service:** internal service to enable admins to ship in-store materials to merchandiser crowd
 - o Performed a full-scale refactor of the shipments service to make it testable and extensible (C#, Dependency Injection, Service Objects, Repos, MongoDB, NSubstitute)
- **Fusion Data Lake:** collecting our company data from various teams into a data sink for future analysis
 - o Daily dump of “retail stores” catalog into sink (MongoExport, MongoDB, Python, Spark, BigQuery)
 - o Cross-cloud stream of store visits into sink (C#, AWS SQS/SNS, Python, Spark, Pub/Sub, BigQuery)
- **Crowd CMS:** admin web service used to manage the legacy crowdsourcing platform
 - o Built the ability to do quick lookups on aggregated submission statistics by date range (Postgres, PL-SQL, DB rollups, DB Views, Elixir, Genserver)
 - o Developed (w/ team) a radically new experience for admins: the ability to use a templated UI to mix-and-match components to generate in-store runbooks (Ruby, Rails, GraphQL, React, Javascript)
- **DMX MVP:** proof-of-concept for web application to assign tasks to direct merchandising workforce
 - o Built the user-registration flow w/ mobile token validation (Elixir, Phoenix, GraphQL, React, Relay)
 - o Built the data models for transactions, work plans, and users/workers (Elixir, Phoenix, Postgres)

Tout
Sr Full Stack Ruby On Rails Engineer

San Francisco
Aug 2017–April 2019

- Designed and built from the ground up a highly configurable video+ads application that can be injected into any VPAID compatible player (Javascript, CSS, Mocha, Chai, Webpack, AWS S3)
- Reverse engineered Brightcove player and built a product that allows publishers to use our tech in conjunction with their existing Brightcove implementation (Rails, Javascript)
- Created adapters for different ad vendors to integrate into our video player
- Developed a new way to load our partner SDK, with an as-needed tiered approach, which dropped average time to screen from 4 seconds to 1 second (Rails, Javascript, Redis, AWS S3)
- Implemented header bidding so that publishers can get the best ad rates from advertisers
- Built various CRUD flows for our partner management dashboard platform (Rails, ReactJS, Mysql)

Ellucian
Software Engineer

San Francisco
May 2015–March 2017

- Built suggestion feed, role assignment, and success score admin sections of advising module (JS, Rails, Pg)
- Developed tenant-specific branding customization page (Rails, Javascript, HTML, CSS, Postgres)
- Integrated third-party student information systems into our main product (Rails, SOAP, XML)
- Built validation service to allow courses to be validated only when being published (Rails)
- Fixed arcane browser quirks and ensured that our site responded to touch devices (Javascript, CSS)
- Created a conditional locking/unlocking functionality for assessments (Rails, Pundit, Postgres)
- Made interactive preview screen for pretests viewable by course designers (AngularJS, Rails, CSS, Pundit)
- Wrote automated tests with Rspec, Capybara, and Siteprism to ensure that features would not regress

BeRecruited
Software Engineer

San Francisco,
March 2014–April 2015

- Launched a popular feature that enabled students to track their favorite universities (ReactJS, Grape)
- Built a multistep activation flow that increased signups over 20% by removing attrition (Rails, HTML)
- Converted site to be responsive (Bootstrap, SCSS, media queries)
- Developed various CRUD forms and flows (Rails, HTML, SimpleForm, AJAX validations)
- Built interactive onboarding dialogs to guide users through new features (jQuery, BootstrapTour)

- Created visually interesting emails that work across platforms (CSS tables, EmailOnAcid, Rails Mailers)
- Implemented coupons and promotions (Recurly, Javascript, Rails, CSS/HTML)
- Integrated video tracking and metrics (Mixpanel, Inspectlet)

Weedmaps

Software Engineer

Irvine, CA / Tucson, AZ

Sept 2013–Jan 2014

- Built entire frontend for Weedmaps Deals, a coupon platform for marijuana dispensaries and patients, which became the company's primary income stream (Javascript, BackboneJS, HTML, CSS, Ruby/Rails)

Infosys

Analyst

San Ramon, CA

June 2007–Oct 2012

- Captained a small team at client (AT&T) to identify and align testing environments for 150+ applications to support integrated systems testing, so that new releases could be tested well and deployed confidently.

EDUCATION

UC Irvine

Bachelor of Science in Psychology (concentration: cognitive sciences)

Irvine, CA

May 2007

SKILLS

- **Languages:** C#, Ruby, Python, Elixir, Javascript, Typescript, Go, C++, SQL, HTML, CSS, Processing
- **Frameworks:** Rails, .Net Core, Phoenix, Express, Nextjs, React, Angular, Tailwind, Bootstrap, FastAPI
- **ML Tools:** PyTorch, FastAI, LlamaIndex, Langchain, TorchServe, RAG w/ vector DBs, MediaPipe, Scikit-Learn
- **Databases:** Postgres, MongoDB, MySQL, AWS RDS, AWS Aurora, GCP BigQuery, Redis
- **Messaging:** AWS SQS, AWS SNS, AWS Kinesis, GCP Pub/Sub, NServicebus, RabbitMq
- **Deployment:** Docker, Terraform, CloudFormation, Github Actions, Jenkins, Linux, Bash, Make
- **Testing:** RSpec, Minitest, Capybara, Siteprism, XUnit, NSubstitute, Moq, FluentAssertions, ExUnit, Mox, unittest, Mocha, Chai, Jasmine, Jest, Storybook, Selenium, Browserstack
- **Design Philosophy:** Grug-brained, DDD, Clean Architecture