Dev Doshi

creative problem solver

Expertise

- understanding business goals, problem spaces, solution spaces, and tradeoffs
- providing technical solutions tailored to the available resources
- applying first principles thinking while considering how things evolved to their current state
- automating systems while understanding which parts need humans in the loop

Education

Carnegie Mellon University
Pittsburgh, PA May 2013

- B.S. Computer Science
- B.S. Human-Computer Interaction

Extracurriculars

- Teaching and Mentoring at organizations like GirlDevelopIt, local bootcamps like Parsity and Project Shift, and UNC Computer Science
- Participating in conferences like React Boston, API World, ASC, RESTfest, All Things Open, and Temporal Replay
- Watching/Listening to podcasts and conference talks.

I'm a lifelong learner who loves to empower others to focus on their actual goals instead of getting sidetracked by accidental complexity. I've touched most parts of the stack, but I've noticed that technology is rarely the most difficult part of solutions. I am excited about working in environments where I get to engage with and learn from the stakeholders and audience, particularly internal consulting, platform development, or skunkworks roles. I have primarily been remote-first my entire career but enjoy collaborating in person.

Experience

Freelance Consultant and Contractor February 2015 - present
Self (https://devdoshi.com) *remote, Durham NC, Atlanta GA*

Typescript · Flutter · Dart · Angular · React · Remix · Nx Monorepo · Postgres · Temporal.io · Python · LaunchDarkly

Some projects from 2022-2023 focused on saving people time and money:

- Created a proof of concept Shopify app for a local print solutions company allowing anyone selling printable digital materials to instantly be able to reach more customers by selling printed copies and not worrying about fulfillment. Leveraged Puppeteer to automate some manual processes and Appsmith to expose backoffice operations.
- Maintained and updated apps and infrastructure/CI/CD/observability for a local medical consult/referral/telemedicine app. Supported migration from native mobile apps, a Python backend, an Angular admin dashboard, and a React PDF generating app to a Flutter/Dart codebase. Deployed on AWS ECS with a mix of Github Actions and Gitlab CI
- Created a production-ready version of a prototype for hourly employee workforce management. Employees can chat with a bot via SMS (Twilio, Dialogflow, Temporal.io) and the bot acts on their behalf to automate scheduling changes and getting in touch with managers. Replaced the Retool prototype with Remix deployed on AWS ECS with Github Actions and Pulumi. Enabled less technical members of the team to affect business workflows without changing application code using Make.com
- Created browser-based automation for a local restaurant group to accelerate and simplify the accounting process by combining data from a variety of sources. Enabled the finance team to experiment with different business models/structures including a Living Wage.

Lead Developer February 2019 - March 2022
Optic (https://useoptic.com) *remote, Durham NC*

Rust · WebAssembly · Node.js · Typescript · React · Yarn Workspaces · Scala.js · Postgres · Kafka · GraphQL

Optic (YCombinator S18) started as an open-source tool to automatically document HTTP APIs by observing live traffic observed from proxies, sidecars, SDK agents, and more. By observing traffic and comparing it to a spec, we were able to generate diffs and suggestions for automatically incorporating these diffs into the spec. As we iterated, we built many tools on top of that foundation. Here's a sampling of some things I worked on in pursuit of making it easier for developers to work with APIs and get the right people involved at the right time:

- Designed and developed a streaming data ingestion pipeline using AWS S3, AWS Lambda, and Apache Kafka deployed with Pulumi via Github Actions. Client agents would upload a representation of traffic and we used that data to track changes, traffic vs. spec coverage, and general performance and error metrics.

↓

Experiments

- Code change-aware checklists / playbooks for pull request review and pattern enforcement. Goal: keep projects documented and aligned with architecture in a sustainable way.
- Annotation-based code generation. Goal: avoid the pitfalls of template-based code generation.
- An append-only object store for code. Goal: find a better way than Git to represent the data we typically interact with in codebases
- automated typescript typing / conversion. Goal: Optic but for code
- content-addressable "file system" for incremental builds. Goal: eliminate the POSIX file system abstraction as a bottleneck
- Github Blocks, WebContainers, and alternative browser based development environments
- LLMs, ChatGPT
- Convex.dev, EventStore, Datomic, SurrealDB, EdgeDB, Graph databases, triple/quad stores, FoundationDB, Velox, and Datalog
- Arc Browser and alternative browsers
- Fig, Nushell, Charm, Warp, Taskfile.dev
- Server Side WebAssembly

- Led the team in designing and developing iterations of: API Spec/Documentation Editors, API SDK and Documentation Generators, Spec Fiddle (a way to exchange and fork API specs), and an API Review/Governance platform
- Made sure our code was modular enough to be used in the browser (with in-memory and backend persistence), in browser extensions, and via NPM (interactively and in CI) so that potential users could try it with as little friction as possible.
- Devised a test automation system oriented around snapshot testing and generating test cases and regression suites from app usage (click a button to dump the current state as a test case or check all existing data against new code). This minimized developer toil and help prevent many categories of bugs while allowing developers to make changes with confidence.
- Devised a JSX component-based code generation framework allowing easy mixing of templates, AST, and other code generation techniques. This was applied to generating API client SDKs, Markdown documentation, YAML OpenAPI documents, code samples, Postman collections, etc. Instead of rendering DOM elements, it rendered a representation of File System artifacts.
- Devised a GraphQL representation of an API specification including data derived from runtime information. The GraphQL schema could be backed by OpenAPI or our event/log-based spec format, and could be extended/evolved using GraphQL Federation.
- Figured out how to inject a proxy into AWS Lambdas so we could automatically document their inputs and outputs by adding a layer.

Software Developer III May 2017 - January 2019

Cengage Learning (https://cengage.com) *remote, Boston MA*

React · Node.js · Typescript · Enzyme · Cypress · react-testing-library · GraphQL (Apollo) · Jenkins · Java (OSGI) · Spring Boot · Pivotal Cloud Foundry · AWS · XMPP · Discourse · TokBox · ElasticSearch · Cassandra · Redis

Cengage is a global educational publisher with multiple online education platforms serving a few million students and teachers.

- Improved the deployment and testing process by creating dashboards and a Slack notification bot that allowed teams to have a shared understanding of the state of the system
- Led UI on a small team implementing a production-ready MVP of discussion forums integrated with Cengage's activity engine that could be provisioned on demand for a given course activity.
- Worked on a larger team improving instructor authoring tools and implementing group video chat for language learning course activities.
- Led UI on a small team implementing an in-application messaging system so the marketing team could target messages to segments of users across all of Cengage's product platforms. This project expanded into a sidebar for Cengage Unlimited that had to work on all of the product platforms.

Software Developer February 2015 - January 2017

Dropsource (https://dropsource.com) *remote, Raleigh NC*

Dropsource is a low/no-code Mobile Application Development Platform built on React/Flux with Node.js and Scala backends and distributed noSQL data stores. Users can drag-and-drop to create their application's interface, add actions to event handlers and connect it to Swagger specifications to make their apps data-driven and interactive, hit our "preview" button to preview their app in the browser via appetize.io, and hit our "build" button to generate human-editable native iOS and Android code.

↓

- Unison Language, Dark Language, val.town, Membrane, Webpack Module Federation
- Remix, Next.js, Solid.js, Crank.js, Blitz.js, etc.
- Cloudflare Durable Objects, R2, etc.
- OCaml, ReasonML, Rescript, Melange
- CloudGraph / AutoCloud

- Dramatically improved the reliability and maintainability of the MVP as I took over its development from the CEO. Added visualization tools to help inspect and debug the app models
- Rewrote the product to make it simpler to develop the more advanced features in the roadmap while improving the performance and reliability. Played a significant role in designing and developing the core internal abstractions and architecture using isomorphic Javascript, React/Flux, graphs, and Event Sourcing

Technical Lead June 2013 - January 2015
Flip Your Training *remote, Raleigh NC*

Flip is an improved version of Classroom Salon built on a cloud-based LAMP stack for commercial use. It facilitates threaded discussions around PDFs, HTML5/Youtube/Vimeo videos, and arbitrary links. Flip also has reply via email, multitenant branding, Stripe payment integration, and analytics tools for administrators. It was designed API-first and supports desktop and mobile web clients.

- Owned development, handled all of the support, and occasionally managed contractors
- Designed and developed a custom Javascript hierarchical routing/data/UI framework
- Designed and developed a novel PDF annotation system and content scrambling

Technical Lead August 2010 - June 2013
Classroom Salon *Pittsburgh, PA*

Salon was a social learning platform built on .NET and Adobe Flex. A major goal was to improve online education for students and teachers. Salon was funded by the National Science Foundation, Gates Foundation through next generation learning innovations grant, Heinz Foundation (through Carnegie Tech. Transfer), Innovation Works and i6 innovation grants.

- Took over maintenance, support, and development of the site and helped manage other students who contributed to the project
- Led the design and development of a major upgrade to the site software, including moving it to a REST API, prototyping alternative discussion UIs, and moving as much Flex as possible to Javascript

↓