

Patrick Hanford

Senior Software Engineer

hanfordpatrick@gmail.com
linkedin.com/in/phanford
github.com/codespent

TECHNICAL SKILLS

Languages: JavaScript, TypeScript, Python, C#, HTML, CSS, Go, GraphQL, Elixir, PHP, Kotlin

Frameworks: React, Vue, NextJS, Django, Flask, React Native, Cordova, FastAPI, Entity Framework, Laravel

UI Libraries: SoftUI, Bootstrap, TailwindCSS, Serenity

Databases: PostgreSQL, SQL Server, MySQL, MongoDB, MemDB

Developer Tools: Git, Docker, TravisCI, Jenkins, GitHub Actions, Android Studio, Azure DevOps, XCode, Prometheus, Grafana, Jenkins

Cloud: Amazon Web Services, Google Cloud Platform, Azure, Vercel, Netlify

Libraries: Pandas, NumPy, Matplotlib, TensorFlow, Tesseract OCR, Linq, ChartsJS

EXPERIENCE

Senior Software Engineer

Robert Half International

Feb 2020 - Mar 2024

- Completed work on >15 projects of diverse tech stacks among >8 clients of varying industry.
- Implemented Docker Swarm for an agency supporting numerous legacy Laravel (PHP) sites.
- Developed a 3-day Docker & Vagrant centric DevOps training module to improve an existing dev team's development pipeline allowing them to turn around client products faster.
- Redesigning a Cordova cross-platform mobile application for React Native, managing various app store compliances and distribution pipelines.
- Developed DotNet Core (C#) API for a mobile application using Entity Framework, Linq, and SQL Server.
- Built series of dashboard views used for manipulating & visualizing outputted data from compared AI models.
- Developed custom OAuth library for internal SSO & identity management.
- Designed a system to maintain data across multiple databases (SQL Server, Postgres, and MySQL) using the Serenity framework (C#).
- Extended and maintained CI/CD pipelines for various app types in Azure DevOps, GitHub Actions, AWS Elastic Beanstalk, and Jenkins.
- Built custom CLI tool using Fire (Python) & CI/CD pipeline used for matching firmware deployments to various environments existing on real hardware.
- Developed a video conferencing application using Django Rest Framework (Python) and NextJS (React).

Software Engineer

StreamKick *Sep 2017 - Sep 2020*

- Handled migration of existing Ruby on Rails codebase to Django.
 - Developed authentication system including OAuth with various providers, middlewares to handle authentication states, & presentation of authentication forms.
 - Built & maintained automation pipelines for various responsibilities such as micro-service releases, tooling distribution, & automated testing.
 - Designed & continuously improved upon multi-faceted algorithms used for matching live-streams to viewers across a plethora of data points.
 - Assumed database administration responsibilities including schema design for Postgres & SQL Server databases hosted on Amazon Web Services (RDS).
 - Implemented Prometheus for resource & app usage metrics presented by Grafana dashboards to aid in scaling appropriately for business needs.
-

MY PROJECTS

DevStreams

A live-stream curation platform that enables viewers to discover & categorize streams based on language, framework, and/or technology tags across Twitch, YouTube, Kick, & Beam.

- Built a series of services tasked with automated stream collection for app consumption & machine-learning model training.
- Implemented OAuth for Twitch, YouTube, Kick, Microsoft, & Google.
- Built a series of services for automatically tagging live-streams at ingress leveraging machine-learning/AI for gradually increasing accuracy & limited manual intervention.
- Built backend dashboard for handling content moderation as well as manual intervention of automated tagging.
- Used OCR (Optical Character Recognition) with TensorFlow models to identify programming languages & frameworks from video thumbnails containing code in view.
- Refactored the frontend codebase from VueJS to NextJS to take advantage of server-side rendering features.
- Implemented GraphQL alongside existing REST API to address over-fetching on creators' profiles.
- Built out a robust performance & usage telemetry system to handle reactive scaling as well as respond to incidents & usage influx.
- Built a series of pipelines for CI/CD, automated testing, data aggregation, etc.
- Built a CLI tool using Python Fire for orchestration of services & quick admin tasks. Later refactored this tool to GoLang to respond to environment concerns causing delays.

Languages: Python, JavaScript, TypeScript, GraphQL, Go, SQL

Databases: Postgres, MongoDB, SQLite

Libraries: React, Vue, NextJS, TensorFlow, Tesseract OCR, Fire CLI, Redis, Docker

Cloud: Amazon Web Services, Google Cloud, Vercel, Netlify

Ops & Testing: Jenkins, Grafana, Prometheus, TravisCI, PyTest, Cypress, Selenium, Jest

Bias

A language/framework agnostic toolkit for reusing common logic & UI components for streamlined development of hobbyist projects & internal tools.

- Built a CLI tool using GoLang that streamlines code generation for bootstrapping projects using a series of guided questions to include starter components, middlewares, & services.
- Built a dashboard using Elixir that allowed easy browsing of available components & services with one-click addition to projects.
- Developed custom middleware templates for logging, authentication, & error handling that could be easily integrated into new projects.

- Implemented a module for real-time updates using WebSockets, allowing components to automatically sync state across instances.
- Designed and implemented a plug-in architecture to support third-party extensions and integrations.
- Developed detailed documentation and example projects to facilitate user onboarding and toolkit adoption.
- Implemented version control for components, allowing developers to lock specific versions or update to the latest stable releases.
- Developed integration with popular IDEs (e.g., Visual Studio Code, IntelliJ) to streamline the development process for users.
- Created scripts for automated deployment and scaling of the toolkit infrastructure on cloud platforms like AWS and Google Cloud.
- Built & maintained documentation & support site with automations integrated to Slack & Discord.
- Created a CI/CD pipeline for the distribution of new releases using GitHub Actions.
- Led beta testing with over 15 developers to drive feature and testing direction.

Languages: Python, JavaScript, TypeScript, Go, Elixir

Databases: Postgres, MongoDB, SQLite

Libraries: Websockets, Docker, VSCode Extensions SDK, IntelliJ SDK, Logrus, Zerolog, Winston, Discord, Slack

Cloud: Amazon Web Services, Google Cloud

Ops & Testing: Jenkins, Grafana, PyTest, Cypress, Jest, Mocha, Chai