

# KATIE JANZEN

Senior Software Engineer (Full-Stack)

---

## QUALIFICATIONS

- Languages: Nix, Rust, JavaScript / TypeScript, Python, Java
- Frameworks: React, Redux, MUI, Node.js, Express, ViteJS, Webpack
- Databases: PostgreSQL, MongoDB / Mongoose
- Testing: Playwright, JUnit, Jest, Puppeteer, Supertest
- Web: RESTful APIs, WebSockets

## CONTACT INFO

- Phone: +1 (806) 654-6933
  - Email: [katiejanzen@347online.me](mailto:katiejanzen@347online.me)
  - Website: <https://347online.me>
  - LinkedIn: <https://www.linkedin.com/in/katie-janzen/>
  - GitHub: <https://github.com/347Online>
- 

## SUMMARY

In one way or another, I've been developing software for just about as long as I can remember. I've built full-stack web applications, video games and game modifications, a command line interface for running a secret santa event, I can do it all. I'm a hard worker so if you need something done right, look no further: I'm your new developer.

## EXPERIENCE

2011 - PRESENT

Senior Software Engineer, Amplify Education  
TODO!

2025 - Present

Front-End Developer, Take Command Health Insurance Agency Inc. 2022 - 2024  
SaaS Company offering Web Platform for Employers to manage a Health Reimbursement Arrangement for their Employees

- Built authorization architecture enforced on both backend event handler and front-end user interface to gate functionality unavailable based on user's permissions
- Automated synchronization of defined backend permissions to frontend removing the need to redundantly modify defined permissions manually
- Developed custom Django quick action to repair damaged company records quickly and accurately
- Built backend rest API for Managing Company Records of Employees deployed to AWS Lambda
- Migrated product team's selected design system build pipeline away from Create React App (deprecated shortly thereafter) to ViteJS delivering superior performance and greatly improved developer experience and iteration time

## OPEN SOURCE

2022 - PRESENT

Contributor, Nix Ecosystem 2024 - Present  
Purely functional package manager, package repository, and software packaging toolset for fully declarative package management and configuration

- Shared system configuration spanning personal workstations, laptops, home servers, and portable Neovim setup
- Contributed package definition to nixpkgs for tsc.nvim Neovim Plugin by dmmulroy
- Contributed declarative module to NixVim for vim-be-good Neovim Plugin by ThePrimeagen

Author, Umpteen 2023 - Present  
Turing-complete Custom Programming Language preferring data immutability bootstrapped from Rust

- Implemented language featureset including conditional and looping control flow, lexically-scoped variables, and first-class functions
- Developed fully-functional Brainf\*ck interpreter in the newly implemented language as sample program

---

## INDEPENDENT VIDEO GAMES

2011 - PRESENT

Coldlight 2020 - 2022

Arcade Action Game where players destroy enemy spacecraft to claim high scores

- Orchestrated a system for a dynamic soundtrack with different tracks being played depending on the game state
- Integrated Juju Adams and Alynne Keith's Input library for seamless control across a wide variety of controller hardware

- Created a complex underlying weapons system shared across the player and enemy objects alike, enabling rapid prototyping of new weapons

A.C.E. - A Chromatic Experience 2016

Puzzle Platformer where players use color to collect keys and traverse various gates to solve puzzles

- Designed level creation suite with intuitive user interface to enable creating new levels quickly, and allow players to create their own
- Implemented a robust control system for keyboard and controller, delivering players the freedom to change controls to fit their needs

Trapfall 2014

One on One Competitive Online Multiplayer Experience

- Built Host / Client architecture enabling players to have multiplayer matches over LAN / internet

FutureShock 2011

Vertical-Scrolling Arcade Shooter

- Integrated Pointed Games web portal API for uploading player high scores to earn 'pointed points'
- 

## PUBLIC PRESENTATIONS

Straight from the SRC: How a Compiler Works 2022

Talk presented as part of SingleSprout's Speaker Series

- Outlined how source code written by developers is converted into functional machine instructions
- Detailed the transformations code goes through in a compiler toolchain from tokens to abstract syntax tree to bytecode/native instructions