

Chase Lane

(412)-995-0352 ❖ lanechase34@outlook.com ❖ github.com/lanechase34

Summary

Full-stack software developer with 3+ years building enterprise web applications for federal clients. Specializes in React/TypeScript frontends paired with ColdFusion/Python backends, with hands-on DevOps experience in cloud providers using Docker and CommandBox.

Experience

Software Developer II, MajorKey Technologies

2025-Present

- Developed and maintained 11 full-stack web applications for the National Park Service, serving as the Subject Matter Expert for four critical systems used by 4000+ federal staff nationwide.
- Re-architected a legacy SOAP service into a RESTful API consumed by each NPS application, focusing on performance, reliability, and new integrations.
- Led a 6-month development effort to build a data aggregation and analytics interface consolidating datasets across NPS and other government bureaus, enabling real-time inline editing and decision-making.
- Designed and implemented CI/CD pipelines using Jenkins and Portainer. Containerized Testing and Production environments and migrated from on-prem infrastructure to Azure, reducing deployment time and failure rates by 90%.
- Migrated legacy application to a structured ColdBox MVC framework, improving maintainability, performance, and enabling modular feature development.
- Led requirement-gathering sessions and client demos, translating stakeholder feedback into actionable tickets and delivering feature walkthroughs that drove official sign-off.
- Established TestBox as the testing framework for a critical NPS application – ensuring code coverage across high-priority endpoints to guarantee reliability in production.
- Authored system specification documents, user manuals, and requirement traceability matrices in compliance with federal project standards.

Software Developer, MajorKey Technologies

2023-2025

- Modernized and upgraded applications from ColdFusion 2018 to ColdFusion 2023, improving functionality, security, and maintainability.
- Resolved critical production incidents across frontend, API, and database layers.
- Actively participated in Agile sprint planning by creating tickets, scoping levels of effort, collaborating with stakeholders, and breaking complex requirements into smaller units.
- Built an internal project management tool integrating Microsoft Graph API and timesheet data to track cost, progress, and development at various steps, producing exportable planning reports.
- Implemented various optimizations including caching, asynchronous loading, and load balancing to improve application speed and responsiveness.

Education

University of Pittsburgh - Bachelor of Computer Science

December 2022

Minors in Statistics, Economics

GPA: 3.6/4.0

Skills

- **Frontend:** React, TypeScript, JavaScript, HTML, CSS
- **Backend:** REST APIs, ColdFusion/Lucee, Python, PostgreSQL, MSSQL, Oracle
- **DevOps:** Docker, Jenkins, CI/CD, Nginx, Azure, AWS, Git, CommandBox
- **Testing:** Jest, React Testing Library, TestBox, Playwright
- **Certifications:** Adobe ColdFusion Certified, Udemy Web Developer Bootcamp

Projects

SpendingTracker | React, TypeScript, Lucee, PostgreSQL

- Designed and built a full-stack expense and subscription tracking SPA using React and TypeScript with a Lucee REST API backend.
- Implemented JWT-based authentication with role-protected routes and conditional UI rendering.
- Built reusable, performant data table components with TypeScript generics supporting pagination, sorting, filtering, and large datasets.
- Handled complex state management using React Context, custom hooks, and Zod for type safety validation.
- Comprehensive test coverage using Jest and React Testing Library for the frontend and TestBox for the backend API.

PogoTracker | Lucee, ORM, PostgreSQL, Bootstrap

- As an avid player of Pokemon Go, I built this passion project to provide advanced analytics into your Pokemon collection, statistics, and medal progress.
- Developed complex data models and reporting views supporting custom user-entered subsets, daily statistics tracking, collection mismatch detection, and historical trend analysis.
- Built responsive UI using Bootstrap with vanilla JavaScript focusing on fetching JSON data for dynamic rendering.
- Nightly jobs using JSoup to scrape, ingest, and normalize multiple sites, updating the Pokemon database.