Jonathan Lee

Seattle, WA, U.S citizen | (206) 823-5507 | jluwcs@gmail.com | linkedin.com/in/jonathanlee1217 | github.com/JonathanLee1217

Summary

Full-stack software engineer with hands-on experience building and deploying production applications using Azure, Python and React. Skilled in test-driven development, CI/CD pipelines, and Agile methodologies. Passionate about writing clean, well-tested code and optimizing system performance. Seeking to contribute to a collaborative engineering team while delivering user-focused solutions.

Education

University of Washington — Seattle, WA **09/2020 - 06/2025**
Bachelor of Science, Computer Science and Software Engineering
Relevant Coursework: Data Structures & Algorithms (C++), Software Engineering, Database Systems, Machine Learning, Operating Systems

Technical Skills

Languages: C++, Python, JavaScript, SQL
Frameworks & Libraries: React.js, Flask, Node.js
Databases: PostgreSQL, Google Sheets API
Tools: Git, GitHub, Visual Studio Code
Cloud & DevOps: Microsoft Azure, Azure DevOps Tools, CI/CD Pipelines, Automated Testing
Operating Systems: Linux (Ubuntu), Windows
Methodologies: Agile, Scrum, Test-Driven Development (TDD), Code Review

Selective Experience

Stock Trend Predictor — Team Lead | Python, LSTM, Azure DevOps **01/2025 - 06/2025**
Led a 4-person team to build a predictive stock analysis platform using LSTM neural networks, delivering 2 weeks ahead of the 5-month timeline through structured Agile sprint planning.

- Architected end-to-end system integrating APIs (stock prices, corporate information, news) with a responsive web interface for real-time stock analysis
- Engineered LSTM model in Python achieving 50% directional accuracy on historical market data; automated training pipeline reduced manual intervention by 20%
- Established CI/CD infrastructure on Azure DevOps, implementing automated testing and deployment workflows that enabled daily releases
- Coordinated cross-functional task allocation and weekly standups, maintaining team velocity and resolving blockers to meet accelerated delivery target

Calendar Services Web Application — Full-Stack Developer | C, Azure DevOps, React.js **09/2024 – 12/2024**
Contributed to a 5-person team enhancing an existing multi-service calendar application, delivering new features and performance optimizations using TDD and Agile methodologies.

- Re-architected front-end service call logic, reducing page load time by 30% and improving dynamic interaction across localhost and deployed environments
- Shipped 3 new calendar services including Gregorian-to-Chinese Lunar conversion, driving 15% increase in user engagement
- Automated file classification system using regex pattern matching, eliminating 4 hours/week of manual tagging
- Redesigned input components for accessibility compliance, achieving 20% improvement in user satisfaction scores
- Managed 10 user stories end-to-end in Azure DevOps; diagnosed and resolved 2 critical bugs affecting user-facing calculations
- Established automated CI/CD pipeline enabling streamlined testing and deployment workflows

Enhanced AI Strategy: Pac-Man — Backend Developer | Python, Pytorch **09/2023 – 12/2023**
Developed enhanced AI strategy for Pac-Man game applying advanced search algorithms, significantly improving gameplay performance in complex scenarios.

- Applied Uniform-Cost Search, A* Search, Minimax, Alpha-Beta Pruning, and Expectimax algorithms
- Improved AI gameplay performance by 30% based on average completion time
- Reduced pathfinding errors by 20% through optimized search algorithm implementation
- Achieved 95% success rate in solving complex game scenarios

FloraHeal: Interactive Plant Care and Disease Management Portal — Full-Stack Developer | PostgreSQL, React.js **09/2022 – 12/2022**
Developed interactive plant care portal in a 5 people team utilizing React and Flask, delivering responsive user experience with optimized data management in PostgreSQL and high-availability Azure deployment.

- Architected single-page application using React, CSS, and Flask API integration
- Triaged and resolved 20 user-reported bugs within 30 days, stabilizing production environment
- Optimized database queries achieving 50ms reduction in average response time
- Implemented CRUD operations boosting data retrieval and modification efficiency by 30%
- Deployed to Microsoft Azure achieving 99.9% uptime for consistent user access