Sean Reed

(519) 319–3097 | reedse.brocku@gmail.com | linkedin.com/in/reed-sean | github.com/reedse

Education

BROCK UNIVERSITY
Bachelor of Science in Computer Science *April 2025*

Experience

Frontend Developer March 2025–Present
Vannoord Property Maintenance *Aylmer, ON*

- Developed a responsive property maintenance landing page using Next.js and React, implementing server-side rendering (SSR) for faster load times and improved SEO.
- Integrated dynamic contact forms with form validation using React Hook Form and Zod, ensuring secure data handling and seamless email notifications via Next.js API routes.

Projects

Quiznator.ca - Study Platform for Students & Teachers: Full-Stack Web App | *React, Next.js, MongoDB*

- Built a scalable backend with Next.js API routes and MongoDB, implementing RESTful endpoints for user authentication (Clerk), quiz creation, class management, and PDF-based text extraction, while enforcing security measures such as rate limiting (Upstash Redis) and input validation (Zod). Secure pro version payments with Stripe.
- Optimized frontend state management and data fetching using React Query and Zustand, implementing client-side caching, optimistic updates, and API polling for real-time quiz progress tracking, reducing redundant network requests and improving performance.
- Developed an interactive and accessible UI with Next.js, Tailwind CSS, integrating dynamic components and custom hooks to enhance user engagement through animated transitions, responsive layouts, and intuitive navigation.
- Designed and executed test cases (unit, integration, end-to-end) to identify and resolve defects across APIs, ensuring code quality and reliability.

Chess with AI Implementation | *Python, PyGame*

- Developed a chess game in Python, utilizing PyGame library, featuring a user-friendly interface and intuitive piece movement for an engaging player experience. With visuals and gameplay like chess.com.
- Implemented an AI opponent using the Minimax algorithm with alpha-beta pruning, allowing it to evaluate board positions and make strategic decisions based on future game states.

Concurrent, Networked Connect Four | *Python*

- Built a multiplayer Connect Four game foundation in Python using Python sockets, establishing reliable communication channels for transmitting game moves and status updates between players.
- Employed threading to manage multiple Connect Four matches simultaneously, allowing players to connect, queue for games, and play concurrently.

Advanced OOP Traffic Simulator | *Java*

- Implemented advanced object-oriented principles, including inheritance, polymorphism, and interfaces, to model vehicles, roads, traffic signals, and other real-world traffic components, resulting in a modular and maintainable codebase.
- Utilized Java threads to simulate the movement of multiple vehicles simultaneously. Employed socket communication for a client/server architecture, enabling real-time traffic updates or remote control of the simulation.

Technical Skills

Languages: C++, Python, Java, JavaScript, SQL, HTML5, CSS3
Frameworks: React, Next.js, Express.js, Node.js,
Developer Tools: Git, Docker, Jest, Cypress, Zod, Postman
Technologies/Frameworks: Redis, Clerk, Stripe, Tailwind CSS, React Query, Zustand, MongoDB, PostgreSQL