Puneet North

Vancouver, BC pkn4@sfu.ca linkedin.com/in/puneet-north github.com/p-north puneetnorth.vercel.app/

Education

Simon Fraser University **Expected Graduation: Apr 2027**
Bachelor of Science, Computer Science *Burnaby, B.C*

- Awarded **BC Achievement Scholarship** for academic excellence.
- Coursework: Algorithms and Data Structures, Intro. to Software Engineering, Computer Networking, Database Systems.

Technical Experience

SFU Computing Science Student Society **May 2025 - Present**
Web System Administrator *Burnaby, B.C*

- Maintained a backend system supporting internal tools and events portal serving a student society of **1,000+** members.
- Optimized **FastAPI** routes and **PostgreSQL** queries to improve response times and resource usage.
- Configured **NGINX** as a reverse proxy to route traffic for test deployment environments, enhancing system modularity and enabling isolated testing.
- Co-led weekly development meetings to plan backend tasks, review PRs, and onboard contributors, improving response time on internal tool issues.

CJSF Radio 90.1 **Apr 2025 - Present**
Software Developer *Burnaby, B.C*

- Automated and led the migration **10,000+** rows of structured data from a legacy **MySQL** database to a new CMS-backed instance, decreasing manual work by **50%.**
- Integrated new Strapi API with existing legacy Drupal infrastructure, reducing content update time by **15%** and simplifying editorial workflows.

Projects

FoodConnect: Food Donation Platform | *React, Typescript, Node.js, Express, PostgreSQL, AWS (S3, RDS), Socket.io, JWT*

- Led a team of 4 as Scrum Master in a semester-long Google (GDSC) hackathon, managing weekly meetings, sprint planning, and task coordination for on-time delivery.
- Built a scalable REST API with **Node.js**, **Express**, and **PostgreSQL**, delivering a fully functional MVP for user authentication, food posts, and messaging.
- Implemented **JWT**-based authentication with hashed passwords and role validation, reducing unauthorized access by over **90**% in testing.
- Deployed backend infrastructure using **AWS RDS** for PostgreSQL and **AWS S3** for image storage, cutting media load times by **25%.**
- Integrated real-time based messaging with **Socket.io**, increasing user engagement by approximately **70%** over traditional messaging.

Mock Interview Minesweeper Game | *Python, FastAPI, MySQL, AWS S3, Google Gemini API*

- Designed a **RESTful API** using **FastAPI** to manage game logic and session states, enabling seamless frontend-backend integration and supporting real-time updates.
- Optimized **MySQL** database queries and schema design to store game progress efficiently, achieving sub-100ms response times for critical operations and ensuring system reliability.
- Configured and integrated **AWS S3**, setting up storage buckets and automating object creation for managing game assets and data.

Mindstream AI | *React, TailwindCSS, Node.js, Express, Firebase, Open AI API, Google Cloud Platform*

- Led a team of 4 in an **Agile** environment to build a platform with **70+** active users that turns lecture notes into interactive study tools.
- Built a backend architecture with **Node.js** and **Express.js**, integrated with **Firebase** for database management, enabling real-time data storage and retrieval.
- Automated **CI/CD** pipelines with **GitHub Actions** and integrated **Jest** testing, reducing release errors by **25%** and accelerating deployment cycles.

AI Text-to-Video Generator | *Next.js, TypeScript, Supabase, Docker, Perplexity AI API, FFmpeg*

- Built an AI-driven content tool that transforms text into short-form video scripts, inspired by social media engagement loops.
- Integrated **Supabase** for secure video and metadata storage, enabling efficient retrieval and playback.
- Containerized deployment with **Docker**, ensuring scalability and reproducibility across environments while reducing deployment times by **50%**.
- Processed videos through **FFmpeg** pipelines, handling input validation, transformation, and rendering via **Next.js** API routes.

Multiplayer Networking Game | *Python, Sockets, TCP/IP, Pygame*

- Designed and implemented with team of 4 a **TCP**-based client-server architecture using Python socket programming.
- Implemented data transmission protocols for player movement and game state synchronization between server and clients.

Technical Skills

Programming Languages: Python, C, C++, JavaScript, TypeScript, SQL, x86, HTML5, CSS
Frameworks & Tools: React, Next.js, Node.js, Express.js, FastAPI, Docker, Postman, Jest, NGINX
Platforms: AWS (EC2, S3, RDS), PostgreSQL, MySQL, MongoDB, Firebase, GitHub, Linux
Concepts: REST APIs, CI/CD (Github Actions), Agile, MVC Architecture, Testing