Philip Chen

philipchen.ca philipchen935@gmail.com philipchen935 philipchen05 647-550-1143

Education

University of Waterloo September 2023 – Present
Bachelor of Computer Science – GPA: 3.9 *Waterloo, ON*

- Courses: Data Structures & Algorithms, Object-Oriented Programming, Reinforcement Learning, Compilers (Adv.)
- Awards: U of T Scholar (**$16.5K**), Software Engineering Scholarship (**$4K**), President's Scholarship of Distinction (**$2K**)

Skills

Languages: Python, C/C++, C#, Java, JavaScript, TypeScript, SQL, HTML/CSS
Technologies: React, Next.js, React Native, Flask, Node.js, .NET, Pandas, NumPy, Redux, Selenium, Linux, Tailwind
Cloud/Database: PostgreSQL, SQL Server, MongoDB, Azure, Google Cloud, Firebase, Vercel, Docker
Spoken Languages: English (Native), **Spanish (Fluent)**, French (Fluent), Chinese (Native)

Experience

Full Stack Developer January 2025 – Present
SparkLease Inc. *Toronto, Canada*

- Building core features **end-to-end** for a multi-tiered full stack application using **C#**, **JavaScript**, and **ASP.NET Core MVC**
- Engineered a large-scale LenderDesk feature from the ground up; designed **12 normalized tables** for **60M+ entries** with automated database updates via WebJob (saving **100+ hours/year**), developed a robust API with **30+ error-handled endpoints** (including comprehensive external documentation), and delivered a clean, responsive UI
- Implemented **18+ pages** and **6+ API endpoints** for a new car discovery feature, driving user activity up by **63%**
- Developed a high-performance data parser using **Python** to flatten and structure data from **20,000+** JSON files for optimizing **SQL** queries on **Microsoft SQL Server**
- Developing a cross-platform **mobile app** using a **React Native**-like framework for expanding accessibility to **150,000+ users**
- Improved code modularity and maintainability by reducing widespread coupling, eliminating **5,000+** redundant lines of code
- Leveraging **CDNs** to reduce load times by **56%** using **Azure Blob Storage** (for images) and **Azure Edge** (for static files)

Software Developer May 2024 – August 2024
Treasury Board Secretariat *Toronto, Canada*

- Innovated a software solution for automating defect reports to save **120+ hours annually** for **70+ developers/QAs** using **Flask**, **Next.js**, and **Firebase**, improving production time by **98%** and report accuracy by **37%**
- Leveraged **Azure DevOps** to push **25+ improvements** to the **CI/CD** of a large data capture solution project, reducing post-deployment incidents by **41%** and leading to a smoother production environment
- Eliminated **10+ hours a week** of manual file management by implementing **PowerShell WebJobs** to manage company file infrastructure, facilitate file distribution/archive disposal, and automate tedious daily tasks

Projects

Memoir | *Python, Flask, MongoDB, React, JavaScript, Tailwind CSS*

- Designed a social media app for sharing and connecting with others through nostalgic memories
- Employed **Auth0** for user **authentication** and **MongoDB Atlas** for secure data storage (posts, user data, networks, etc.)
- Features a **dynamic network graph** with **100+ nodes** for illustrating user-post connections; data is semantically analyzed with **Cohere** and processed through a **BIRCH Clustering** algorithm from **scikit-learn** before visually rendering with **D3.js**

Students of Watan (Settlers of Catan) | *C++*

- Developed an **object-oriented** strategy game in C++ with **20+** modularized classes and **2,500+** lines of code, applying best practices for **scalability** and maintainability
- Implemented Observer, Decorator, Factory, and Strategy **design patterns** to enhance code **modularity**
- Optimized memory management through careful handling of raw and smart pointers, ensuring **0 memory leaks**
- Designed **UML diagrams** and detailed documentation to enforce **loose coupling** and **high cohesion**

Zombie Survival | *Java, Swing*

- Created a top-down zombie survival video game in **Java** using **object-oriented programming**
- Employed **multithreading** to concurrently execute **20+** mutually interacting objects, including $Zombie$ objects, $Player$ input, and **state mutation** (timer, player health, etc.); asynchronously executes and manages **1,000+ threads** in later rounds
- Features a dynamic UI implemented with **Swing** and **Java AWT**; hosted with **JFrame**