

Samuel Gibson

360-454-8190 | sdgibson18@outlook.com | Arlington, WA | <https://www.samuel-gibson.com>

Education

Bachelor of Science in Software Engineering

August 2021 – December 2023

Washington State University, Everett, WA

Technical Skills

Programming Languages: C#, SQL, Java, Python, Javascript/Typescript, HTML/CSS, MATLAB.

Frameworks: Blazor, .NET, JUnit, NUnit, WinForms, Godot, NodeJS, Express, React, Redux, RESTful APIs.

Developer Tools & Platforms: Azure Cloud, Gradle, Checkstyle, GitHub, GitLab, Neo4j, GitHub Actions.

Experience

Volunteer Software Engineer (C#)

March 2024 – June 2024

Remote, WA

- Developed a static web application for upload, retrieval, and management of images, image tags, and text with Azure Static Web App, SQL Database, Azure Blob Storage, and Azure Functions in Blazor, improving the client's outreach, productivity, and reducing service costs by 33%.
- Utilized GitHub OAuth to prevent anonymous REST API calls securing client's backend resources.
- Maintained the stability of application by debugging and updating dependencies, fulfilling client requests.

Data Structures and Algorithms Teaching Assistant

January 2023 – May 2023

Washington State University, Everett, WA

- Conducted Zoom office hours twice a week, verbally teaching students programming, algorithm design, environment setup, object-oriented principles, and troubleshooting technical issues, improving class success by 25%.
- Graded assignments in an online class of 20, allowing the professor increased productivity.

Projects

BanWho? Data Analytic Web App (C#)

January 2024 – March 2024

- Independently developed a full stack web application to discover and display statistics from raw data retrieved by the Riot Games API with Azure App Service, SQL Database, Entity Framework Core, and Blazor.
- Coded algorithms for efficiently gathering, crawling, aggregating, and storing of over 100,000 data entries at 60 minute scheduled intervals, resulting in accurate, up to date, and comprehensible data presented to users.
- Utilized the clean architecture pattern and dependency injection to improve maintainability of the application.

Bullet Hell Game (C#)

January 2023 – May 2023

- Collaborating as a team, applied software design patterns (observer, command, strategy, singleton, composite, flyweight) to create a scalable and extendable 2D game, optimizing the rate of implementing new features.
- Employed design principles such as encapsulation, coupling, open-closed, and substitution, improving maintainability, code quality, and separation of concern.

Checkstyle Plugin (Java)

August 2022 – December 2022

- Independently developed a plugin that checks additional metrics with Eclipse Checkstyle including 5 Halstead metrics: difficulty, effort, length, vocabulary, and volume to further enforce maintainability and quality of code.
- Tested plugin with white, black box, and unit test cases focusing on branch, statement, and fault coverage.

Boeing Scholars Bolt Preload Analysis (Python)

August 2022 – May 2023

- Elicited and analyzed software specifications by regularly communicating with stakeholders over 8 months.
- Documented architecture, systems, and logic for the application both in technical papers and code comments.
- Led software development in an agile multi-disciplinary team to visualize experiment data with Python Tkinter, enabling clear presentation of the team's results, landing the team in the WSU Business Competition finals.