

# AYESHA ARSHAD

Whitby, Ontario | 647-963-9031 | ayesharshad88@gmail.com

LinkedIn: <https://www.linkedin.com/in/ayesha-arshad-8332531b2> GitHub: <https://github.com/ayesha133>

Portfolio: Ayesha Arshad | Portfolio ([ayesha-arshad.duckdns.org](https://ayesha-arshad.duckdns.org))

## EDUCATION

**Bachelor of Engineering: Electrical Engineering | McMaster University | Hamilton, ON**

**2020-2024**

## EXPERIENCE

**Associate Application Developer | ADP Canada | Toronto, ON**

**July 2024 – Present**

- Developed and maintained RESTful APIs for a large-scale Leave & Absence application using **Java** and **SpringBoot**, ensuring seamless functionality for ADP's clients in managing various time-off requests for their employees.
- Managed **PostgreSQL** databases, including writing and optimizing complex **SQL** queries to fetch, read, and write data across multiple databases, supporting high scalability and reliability.
- Maintained service-oriented architecture and participated in the design and optimization of backend systems for enhanced performance and efficiency.
- Collaborated with product managers to assess requirements and implement new features for a global user base.
- Ensured high code quality through unit, integration, and end-to-end testing and followed Domain-Driven Design (DDD) principles to create modular and maintainable systems.

**Software Engineer Intern | Microsoft | Mountain View, CA, USA**

**May 2023 – August 2023**

- Designed and built an API for on-call engineers to help assess the criticality of customer tickets for service downtime using **ASP.NET** and **C#**, optimizing incident response by **50%**.
- Led collaborative efforts with Product Managers to ensure seamless alignment between API implementation and business requirements in a fast-paced environment.
- Deployed the API on **Azure**, integrated **Telemetry** to monitor API performance and latency.
- Enriched code quality and reliability by implementing **MSTest** and **Moq** for thorough API unit testing.
- Developed a **Blazor** web application that displays the criticality of an on-call engineer's customer tickets using the API and **KQL** database queries.
- Optimized web app functionality by proactively implementing user engagement, exceeding MVP expectations and improving performance by **80%**.

**Explore Intern | Microsoft | Redmond, WA, USA**

**June 2022 – September 2022**

- Built a questionnaire widget for the MSN Money Team that provides users with personalized financial recommendations with **React**, **TypeScript** and APIs.
- Created **Telemetry** dashboards to collect metrics and monitor user engagement of the project while in flight.
- Flighted the project to **5%** of users, about **1.25M** daily active users and integrated the project into the MSN Money codebase.
- Initiated performance enhancements for an MSN Money feature, leading to a **30%** improvement in performance.

**Production Engineering Fellow | Major League Hacking**

**May 2021 – August 2021**

- Built, containerized, and deployed a scalable Flask web app integrated with a **PostgreSQL** database on **AWS EC2 instance**.
- Developed a **CI/CD pipeline** for the website through Github Actions for automated testing and live deployment.
- Configured Linux systems and infrastructures.

## SKILLS

- Languages: Python, Java, JavaScript, TypeScript, C, C++, C#
- Technologies: HTML, CSS, React Native, React, Flask, Docker, PostgreSQL, AWS, Firebase, Android Studio, Linux, SQL, Figma, .NET, Blazor, Azure, MSTest, Git, Microservices, Node.js, Agile, Splunk, SpringBoot, Concourse CI, SonarQube, Cucumber.

## PROJECTS

**StreetEats**

**August 2021**

- Built a **Flask** web application with a **Python backend** that suggests restaurants based off user location using an API and IP networking. Allows users to login/register and add/delete restaurants to user created lists using a **PostgreSQL** database.
- Deployed on an **AWS EC2 instance** with an **NginX reverse proxy**.
- Integrated **CI/CD** pipeline to allow automated testing and deployment, **cAdvisor** for monitoring, and **Docker** for containerization.