**Anthony Kilde**

414 26th Street NW Apartment 1, Bemidji, MN 56601  
(218) 275-5652 | [anthonykilde@outlook.com](mailto:anthonykilde@outlook.com) | <https://www.linkedin.com/in/anthonykilde/>

**SUMMARY STATEMENT**

Computer Science graduate with a solid foundation in Object-Oriented Programming (OOP) with interests in Full Stack .NET development, application development, and software engineering. Proven ability to create immersive solutions and drive continuous learning through hands-on projects. A proactive collaborator and problem-solver ready to apply skills in a professional setting.  
  
**EDUCATION Bachelor of Science in Computer Science**, Bemidji State University, Bemidji, MN

**SKILLS** C++, C#, Python, Java, JavaScript, Typescript, HTML, CSS, VB, ASP.NET, Azure, Cosmos, Blob, React, Next.js, Bootstrap, Swift, MySQL, JSON, Visual Studio, Unreal Engine, Xcode

**WORK EXPERIENCE Full-stack Developer**,January 2023 – present

**OmniReality LLC,** Bemidji, MN

* Developed VR experiences utilizing Unreal Engine, Visual Studio, and C++.
* Implemented custom algorithms to control physics and interactions.
* Implemented graphical systems that aided the designers’ workflow.
* Utilized JavaScript, React, Next.js, CSS, HTML, ASP.NET, C#, JSON, Bootstrap, and Azure Web Apps to deploy and maintain web applications.

**ACHIEVEMENTS Outstanding Student in Computer Science**, March 2021, April 2023Bemidji State University, Bemidji, MN

**Presidential Honor Roll**, December 2019

Bemidji State University, Bemidji, MN

**PROJECTS Web Portfolio – (NextJS, React, .NET, C#, TSX , CSS, HTML, Cosmos, Blob, SAS token)**

* **Dynamic Rendering & Singleton Service**: Developed a component-based architecture in Next.js/React with TypeScript, facilitating parallel data loading and caching through a singleton service.
* **Modular CSS and UI**: Leveraged a modular CSS approach to convert high-fidelity wireframes into interactive UI.
* **Data Layer & Backend Service**: Integrated Azure's Cosmos DB and Blob Storage for scalability, deploying a secure .NET backend service for interaction with Cosmos DB and retrieval of Blob links.
* **Security Measures**: Managed secure direct blob access and cross-origin requests using short-lived Shared Access Signatures (SAS) and CORS.

**Apprise (iOS App)**

* **Project Lead**: Led the development of "Apprise," a real-time academic communication app.
* **User Interface**: Collaborated with designers to develop an intuitive and familiar interface.
* **Security Measures**: Implemented anti-cheating and user privacy measures, ensuring secure data transmission.
* **Firebase & JSON**: Integrated Firebase for real-time updates and offline functionality, managing data as JSON.
* **Dynamic Invites & Access Control**: Enabled dynamic classroom invites and provided role-based access control.

**PROJECTS** **1974 Seeburg Quadraphonic Sound Digital Jukebox VR (Unreal Engine, C++, Python)**

* **Python Web Scraper**: Developed a scraper for song integration.
* **C++ Mechanics**: Utilized C++ for jukebox's operational components.
* **Cross-Language Integration**: Bridged Python and C++ for data handling.
* **VR Packaging**: Packaged application for Oculus compatibility.

**Not The North Pole VR (Unreal Engine Game, C++, 3D Math, System Architecture)**

* **Encapsulation**: Created the InstancedSpawnLocationManager class, encapsulating related data and methods.
* **Inheritance**: Used inheritance in game classes for diverse enemy behaviors.
* **Constructors/Destructors**: Utilized these methods to manage resources effectively, improving game performance.
* **Polymorphism**: Implemented polymorphic functions in enemy behavior and spawning.
* **Abstraction**: Simplified complex operations, like enemy spawning, through method abstraction.
* **Error Handling**: Integrated error handling to manage potential issues, improving game stability.

**Python Web Scraper – Kayak.com**

* **Web Scraping & Automation**: Crafted automated Selenium Python scraper for flight data extraction from Kayak.com, with 4-hour interval schedules.
* **Cross-Platform Compatibility**: Established dynamic driver setup for Chromium and Chrome, guaranteeing scraper's OS versatility.
* **Data Management and Error Handling**: Streamlined data extraction with XPath and regex. Implemented error handling to maintain data integrity.
* **CSV Output**: Utilized Python's csv library for automating data output into CSV files, enabling easy analysis and manipulation.

**Python Web Scraper – Youtube.com**

* **Automation**: Automated scraping YouTube, downloading, and audio conversion process, streamlining batch operations.
* **Web Scraping**: Used Selenium WebDriver for browser automation, efficiently fetching user-specified songs from YouTube.
* **User Interface**: Integrated the tkinter library for file and directory dialogues, enhancing user interaction with the system.
* **Exception Handling**: Implemented robust error handling and utilized explicit waits to ensure stability and speed of web scraping operations.
* **Browser Automation Options**: Added a headless browser operation feature, improving efficiency and server compatibility.
* **File Handling**: Developed functionality to manage song lists using .txt files, with the system effectively reading and interpreting the provided song lists
* **Audio Conversion**: Utilized ffmpeg and youtube-dl libraries to download/convert YouTube videos into WAV format.