

**Darren Bentler**  
[ny.dbentler@gmail.com](mailto:ny.dbentler@gmail.com)  
+1 (631) 408-1093

<https://www.linkedin.com/in/dbentler>  
South Huntington, New York (open to relocation)  
<https://www.darrenbentler.com/>

## **Education**

**New York University** | New York City, New York

**Completed:** May 2021

*Studied for Bachelor's of Science in Computer Science and Engineering*

## **Skills**

### **Programming Languages**

- **C#:** 2 years of hands-on experience developing enterprise websites and REST APIs utilizing tools and frameworks such as .NET, Entity Framework, and Roslyn.
- **JavaScript/TypeScript:** 2 years of hands-on experience alongside 3 years education experience building dynamic frontends for enterprise websites using frameworks such as React, Angular, Bootstrap, Material, and Aurelia.
- **Python:** 4 years of experience creating web-automation scripts using technologies such as Selenium, and data visualizations with Matplotlib.
- **C/C++:** 2 years of educational experience with strong emphasis on data structures, algorithms, and operating systems.

### **Software & Tools**

Git, MySQL, SQL, Eclipse, Oracle, Visual Studio, Visual Studio Code, Docker, Azure, Google Cloud Platform

## **Experience**

**biBERK Business Insurance** | Wilkes-Barre, Pennsylvania

**Full Stack Software Engineer (Remote)** | August 2022 - Present

- Crafted and implemented a data-driven solution for the validation of Worker Compensation policies utilizing Angular, C# .NET, and Entity Framework, ensuring seamless compliance with a diverse set of State laws.
- Pioneered the development of an intuitive, in-house web-based configuration tool leveraging Azure Cosmos DB, C# .NET, Azure Functions, and Angular - empowering product owners, business analysts, and developers to efficiently manage underwriting questions with integrated validation for embedded logical expressions and conditionals.
- Revitalized and optimized critical SQL stored procedures, slashing execution times from multiple minutes to mere seconds, which resulted in significantly elevated producer and customer satisfaction during the policy purchasing process.

**Dice Habit** | Los Angeles, California

**Full Stack Developer (Remote, Part-time)** | May 2022 - November 2022

- Orchestrated the development of a tailored website allowing retailers to effortlessly access and customize specialty dice orders.
- Leveraged React's component-based architecture to create a modular and scalable solution tailored to client specifications.
- Oversaw the deployment of both the website and MySQL database on the Google App Engine via the Google Cloud Platform.

## **Personal Projects**

<https://github.com/dbentler>

### **Personal Portfolio Website**

*Languages and Tools: React, TypeScript, HTML, CSS*

*Deployed on the Google App Engine*

- Designed and built a website from scratch using the aforementioned technologies to act as an online résumé/portfolio hub.
- <https://www.darrenbentler.com/>

### **Blamo - A Wolfenstein-style 3D Raycast Renderer**

*Languages and Tools: C, SDL2, CMake*

- A software renderer written in C that morphs a simple 2D array into a dynamic 3D world.
  - Supports variations in wall color depending on the index value in the 2D array (ie: 1 for Red, 4 for Green).
  - Other features include: camera collision, camera rotation on the X and Y planes, and strafing.

### **LW-Network | LW-Core, LW-Hub, PunishGUI**

*Languages and tools: Java | Spigot API | Linux | Tebex API | Bungeecord*

- Collaborated with team members to create a network of game servers and to later develop software to enhance the overall user experience.
- Solved numerous problems such as:
  - Figuring out a point's position within 3D space and its distance from the origin.
  - Resetting a player's gamestate with the execution of a command.
  - Developed and deployed a *monetization scheme (store)* via Tebex's API.
  - Interfacing with the gamestate itself via Spigot API to pull data such as weather, heat zones, and player points of interests for later review.
  - An easy to use GUI system to deal out the appropriate punishment for rulebreakers, instead of relying on memorization and procedure.
  - Ensuring that punished players could not connect to the network on an alternative account, or if they were bypassing an IP ban via a VPN.

### **Silicoin - A C++ Blockchain Implementation**

*Using C++*

- Set out to better understand Blockchain technology by writing my own "coin".
  - Set down the building blocks by creating a data structure - "Block" - which stored an index number, a proof number, previous hash, transaction data, and timestamp.
  - Utilized a C++ implementation of the SHA256 algorithm in order to hash the block data and implement a "proof of work" CPU algorithm for creating more blocks.
  - Chained these implementations together in order to create the "blockchain".