

# STEVEN SCHIFFNER

250 Ridgedale Ave ♦ Florham Park, NJ 07932  
(973) 309-2046 ♦ stevenschiffner3@gmail.com

## OBJECTIVE

---

I am a recent graduate with a background in mathematics and a passion for web development. I am a problem solver and a self-starter looking for an entry-level position where I can continue to develop my skills and add value to the organization.

## TRAINING

---

### **Rutgers Data Science Bootcamp**

Worked in teams to create a series of projects which developed my skills in technologies including but not limited to: Excel, Python (Pandas/Matplotlib), the ETL process, SQL (including pgAdmin), Tableau, and machine learning.

## EDUCATION

---

### **New York University**

BA in Mathematics

Minor in Physics

*Relevant Coursework:* Introduction to Computer Science, Web Development and Programming, Calculus II, Calculus III, Linear Algebra, Discrete Mathematics, Theory of Probability, Mathematical Statistics, Honors Analysis I+II, Honors Algebra I+II

## TECHNICAL STRENGTHS

---

### **Programming Languages**

JavaScript, HTML, CSS, Python, SQL

### **Software & Tools**

Excel, React, node.js, Express, Jest, pgAdmin, Linux

## PROJECTS

---

### **NYCMelee Stats** • April 2025 • [Live](#) • [Code](#)

- Created a website using React which allows competitors in Smash Bros. tournaments to look up their stats
- Queried data from start.gg's GraphQL API, then stored it in a PostgreSQL database
- Displays players' recent tournament results, graphs depicting their statistics over time, and a "back of the baseball card" player card

### **CDC Data Analysis** • Jun 2024 • [Live](#) • [Code](#)

- Cleaned CDC data set with Pandas
- Stored resulting tables in PostgreSQL
- Created dynamic visualizations with JavaScript including a choropleth map with Leaflet, and charts with Plotly

### **Battleship** • Apr 2023 • [Live](#) • [Code](#)

- Re-created the popular board game 'Battleship' with a computer opponent
- Implemented game logic via object-oriented programming
- Used Jest to write unit tests for custom class methods and game helper functions