

# ZACH BOEMER

682-472-0811 ◇ zacharyboemer@gmail.com

github.com/zachboemer ◇ boemer.dev

## EDUCATION

---

**Texas State University, San Marcos**

*August 2018 - August 2021*

Bachelor of Science in Computer Science

**Texas A&M University, College Station**

*July 2016 - May 2018*

## TECHNICAL SKILLS

---

**Languages & Frameworks**

PHP, C#, Magento, Java, SQL, JavaScript, C++, Python

**Software & Tools**

Git, Jira, JetBrains, Visual Studio

## WORK EXPERIENCE

---

**CJPonyParts, Harrisburg, PA**

March 2022 - April 2025

*Backend Web Developer*

- Actively participated in code reviews and QA processes for fellow engineers, providing constructive feedback and ensuring adherence to best practices
- Integrated RESTful APIs to facilitate seamless communication with third-party services
- Developed an innovative solution to obfuscate 3D files on the site, significantly reducing the risk of unauthorized access and theft

**ShoppingGives, Chicago, IL**

December 2021 - May 2022

*Jr Software Engineer*

- Responsible for designing, building, and maintaining back-end services such as APIs and databases for the e-commerce platform
- Implemented micro-services from the ground up aligning with CQRS and event-sourcing methodologies
- Reduced query times by 80% on high-frequency requests
- Actively participated in grooming, pointing, and planning meetings as a part of the agile sprint process

## PROJECTS

---

**Spotify Morning Show - Python, SQLite, GPT-4o, External APIs**

Built a Python-based system that curates a personalized "morning show" playlist by blending Spotify tracks and recent podcast episodes from RSS feeds. Integrated the Spotify API for dynamic content retrieval and used a scheduling algorithm to alternate songs and podcasts within a target duration. Leveraged ChatGPT for podcast summaries, and deployed webhooks to directly notify users.

**News Aggregator - Python, Flask, React, REST APIs**

Developed a full-stack web application that aggregates and filters news articles from an external API. Implemented a RESTful API to handle communication between the React frontend and Flask backend. This project also utilizes axios for making API calls and uses React Router for client-side routing.

**Quadtree Image Generator - Python**

Developed a python program that receives the name of a .pgma file and a variance threshold as inputs, then recursively divides the image into quadrants while the variance of the quadrant exceeds the threshold. The program would then save the new image in the local directory.