

## Relative Links:

Github: [Github Link](#)

Linkedin: [Linkedin Link](#)

Portfolio: [Portfolio Link](#)

# David McNeill ♦ Full-Stack Developer

Mechanicsville, VA 23111 ♦  (804)-683-7779 ♦  [dev.dmcneill@gmail.com](mailto:dev.dmcneill@gmail.com)

## Skills:

### **Back-end Frameworks:**

.NET Core(ASP.NET Core MVC/Web API) ♦ Django.

### **Front-end Frameworks:**

Angular ♦ Vue.js ♦ React.

### **ORM's:** Entity Framework

♦ SQLAlchemy.

### **Other:** PostgreSQL ♦

Heroku ♦ MySQL ♦ SQLite ♦ Chrome Extension ♦ Unity ♦ Node.js ♦ HTML ♦ CSS ♦ SCSS ♦ SASS.

### **Languages:** C# ♦ Python

♦ Golang ♦ JS ♦ Typescript

## Education:

### **CareerFoundry:** Python

For Web Developers  
Advanced Bootcamp.

## Other Interests:

**Outlier:** Contributor for OutlierAI, supporting the development of large language models. Responsibilities included evaluating and refining AI-generated responses based on user prompts and qualitative criteria.

**Unity:** Worked in the Unity Editor for game development, utilizing animations, canvas layouts, game objects, and C# scripts.

## Software - Work Experience:

### **Bleached Solutions ♦ Full-Stack Developer ♦ March 2021 - Jan. 2025**

#### Live Music Archiver - Bleached Solutions Project

- ❖ Capitalized on the latest Chrome Extension version, to seamlessly organize script injections into windows/tabs. Employed webpack/React to bundle the extension. Fashioned six interactive SVGs(with buttons/inputs) to enhance the UI/UX.
- ❖ Engineered 20 utilities for customized data scraping by the user selecting page elements, providing a template for automated data extraction on the backend (Golang - Colly library).
- ❖ Streamlined client scraping .NET and Typescript APIs, cutting roughly 200 lines of code per scraper by removing redundancies and adopting a modular design.

#### APPO Retreats - Bleached Solutions Project

- ❖ Contributed to the development of a fully reactive web app using Angular, ensuring a seamless and dynamic user experience. Designed and implemented seven distinct UI/UX layouts tailored to key application routes, optimizing navigation and usability. Leveraged Angular Material to enhance visual consistency and interactivity, creating a polished and intuitive interface across devices.
- ❖ Worked with observable functions to stream data, based on search keywords. Customized Golang API queries based on search requirements, and lessened unnecessary data fetching for the pagination process by about 50%.
- ❖ Configured metadata and tags for SEO in Google Search. Optimized the site to improve indexing, often achieving first-page results for common searches.

#### Bleached Solutions - Bleached Solutions Main Website

- ❖ Optimized Vue.js component interaction using computed properties and props. Refactored repetitive UI logic into modular components, cutting code volume by ~25% and improving maintainability. Supported .NET APIs integrated with PostgreSQL via Entity Framework.
- ❖ Designed and stylized a highly customizable UI in Vue.js, incorporating a dynamic dark/light mode toggle and offering 15 color options for buttons, inputs, and other UI elements. Leveraged props to seamlessly pass and apply these color preferences across components, ensuring a consistent and personalized user experience. This approach allowed for quick and easy customization of UI elements, enhancing both accessibility and visual appeal.
- ❖ Helped develop a Node.js project to make requests to various sources for piping in data. Utilized GCP services, such as Google Sheets API and Cloud Functions, to input data into Google Sheets and automate workflows.

### **Revenue IQ ♦ Intern ♦ September 2020 ♦ March 2021**

- ❖ Led the refactoring of approximately 25 microservices, significantly enhancing code quality and readability. Implemented the repository pattern within the microservices, ensuring layers communicated through interface contracts (worked with ASP.NET Core Web APIs). This approach helped adhere to microservice architecture principles by separating REST API calls from the business logic meant for isolated tasks.
- ❖ Established testing environments with Moq for the microservices and identified and resolved an average of four bugs per microservice.
- ❖ Overall, my refactoring reduced redundant code per microservice by an average of approximately 400 lines.