DANE WILLIAMS

dwilli36@nd.edu | danewilliams.me | 425 988 4749
linkedin.com/in/danerwilliams | github.com/danerwilliams

Education

University of Notre Dame — Notre Dame, IN
***B.S. Computer Science**, College of Engineering* — *Aug. 2018 - May 2022*

- Cumulative GPA: 3.73/4.0, Major GPA: 3.86/4.0, Dean's List (Spring 2020, Fall 2020)
- Relevant Coursework: Data Structures, Compilers, Systems, Discrete Math, Logic Design, Computer Architecture, Theory of Computing
- Notre Dame Water Polo: Great Lakes Conference champion, elected team officer

Eastside Catholic School — Sammamish, WA
Salutatorian — *Aug. 2014 - May. 2018*

Experience

Notre Dame Visualization Lab — Notre Dame, IN
Data Visualization Researcher — *Aug 2020 - Present*

- Researcher under Dr. Chaoli Wang working with D3.js javascript data visualization libraries
- Designing and building visual analytical tools of Notre Dame student learning data with the goal of helping close academic performance gaps which result from disparities in student backgrounds

AT&T — San Ramon, CA
Technology Development Program Intern — *Jun 2020 - Aug 2020*

- Contributed to AT&T Labs research in detecting 5G network anomalies:
  - Developed a Python application for centralizing virtual network function data with an Apache Pulsar message bus (similar to Kafka)
  - Processed data with Python and Apache Flink data stream computation engine (similar to Hadoop and Spark)
- Scrum master and full stack developer among a team of interns which built a MERN stack (Mongo, Express, React, Node) application for improving student engagement in online classroom environments

Symetra — Bellevue, WA
Cloud Intern — *May 2019 - Aug 2019*

- Gathered and visualized over 20,000 network infrastructure related data points by writing various Python scripts which drove strategy leading to over $50k in monthly circuit savings
- Automated invoice processing with Python, Boto3 SDK, AWS Machine Learning: Textract (Optical Character Recognition) and Comprehend (Natural Language Processing), as well as other AWS services: SQS, SNS, and S3

Projects

- **Tmux Dracula Theme:** Open source maintainer and contributor for the official Dracula Theme, github.com/dracula/tmux
  - Wrote the original codebase for a tmux (terminal multiplexor) plugin which was accepted to the official Dracula Theme.
  - Use API's and Unix pipelines with shell scripts to gather various pieces of system information and display in the tmux status bar.
  - Commit code, merge pull requests, perform code reviews, and manage issues (bugs, feature requests, todos) for the project.
  - Project has over 100 github stars, averages hundreds of clones per week, and has attracted several contributors from around the world.
- **Pastebin and URL Shortener:** Site for shortening URL's or uploading small files built in serverless AWS, github.com/danerwilliams/pstb.in
  - Used the Chalice framework for Python to implement AWS Lambda, S3, and API Gateway microservices as a backend.
  - Developed a simple frontend client in HTML, CSS, and JavaScript which is deployed to pstb.in using AWS CloudFront.
  - Implemented a continuous integration pipeline with Github actions to automatically deploy client to AWS when changes are merged.
- **GroupMe Chat Bot:** An AI chat bot written in Python, github.com/danerwilliams/pork-chop
  - Developed a GroupMe chat bot in Python that participates in conversation as well as responds to custom command modules.
  - Runs on a Flask server and can be trained for conversation from a custom csv dataset or other corpus files using the ChatterBot module.
- **B-Minor Compiler:** A 4 stage compiler for the B-Minor programming language consisting of a scanner, parser, formatter, and typechecker
  - Scans B-Minor source code for legal tokens using regular expressions written in Flex (Lex).
  - Parses tokens for proper syntax using a context free grammar written in Bison (Yacc).
  - Builds an abstract syntax tree in C from the parsed tokens, and then prints the source code using a consistent and readable format.
  - Resolves variables and then checks for legal type assignments in the abstract syntax tree to ensure assembly code can be generated.

Certifications

- **The University of Texas at Austin:** Think Before You Design Think
  - UT Austin coursework offered at AT&T on utilizing design thinking principles to develop customer centered engineering and business solutions

Skills

Languages: Python, C, Shell, C++, JavaScript, Java, Verilog, HTML/CSS, ARM Assembly

Technologies: Linux/Unix, AWS, Git