Bavanan Bramillan
Email: bob.bramillan@gmail.com | LinkedIn: linkedin.com/in/bavananb | GitHub: github.com/bobbramillan/bobbramillan | Site: sitey-lovat.vercel.app/

Education

Drexel University – Philadelphia, PA | ***B.S.*** **in Software Engineering (Expected June 2027)**

- **Relevant Coursework**: Data Structures & Algorithms, Software Requirements, Linear Algebra, Database Management, Systems Programming, Project Management, Discrete Math

Experience

SRI International – Princeton, NJ | 04/2025 – 09/2025 | Software Engineer Intern

- Report Automation (LiDAR Data Reporting System)
  - Problem Context
    - Engineers relied on multiple internal tools to process binary LiDAR sensor data and manually generate plots in Excel, a time-consuming and error-prone process
    - Report preparation took up to one week per device, consuming significant engineering time
  - Solution Approach
    - Unified existing tools into a single **PyQt5 GUI** application capable of batch-processing binary files using **NumPy** and **generating PowerPoint/Excel** reports automatically
    - Collaborated with Test Engineers to design the GUI and **automate key LiDAR calculations** (e.g., Holdoff Time, Angle of Incidence)
    - Implemented a **JSON-based** configuration system and automated report creation via **PyWin32**, dynamically inserting **metadata-driven charts, titles, and images**
    - Packaged the tool as a standalone executable with **PyInstaller** and integrated **GitLab CI/CD** pipelines for automated builds and versioned distribution
  - Impact
    - Reduced report generation time from **1 week to about 1 hour** and standardized report formatting across devices
    - Documented approach allowing future maintainers to evolve, replicate, and integrate GUI into other internal tools
- Manufacturing Database Website (MIDAS)
  - Problem Context
    - Engineers, technicians, and project managers tracked builds, assemblies, tests, and customer data across massive Excel workbooks and paper travelers, leading to frequent mismatches and poor scalability
  - Solution Approach
    - Designed a **PostgreSQL** backend with normalized relational tables, using **unique device IDs to link** assemblies, reports, parts, tests, and customers
    - Created **custom API routes** in NextJS for schema-safe SQL queries, secure updates, and **full CRUD** functionality
    - Developed a **React/NextJS** frontend using **shadcn + Tailwind CSS**, supporting search filtering, single-device views, part-number drill-downs, and real-time throughput/defect metrics
    - Integrated with Empower Omnify PLM via **mutual TLS authentication** (server root + user PFX certificates)
    - Deployed internally using **Nginx reverse proxy and IIS**, providing access via hostname
  - Impact
    - Replaced static Excel tracking with a multi-user, web-based system, improving data integrity, collaboration, and production scalability
    - Documented approach for future maintainability and Quality Assurance

Beats By Dre – Remote | 06/2025 – 08/2025 | Data Analyst Extern

- Collaborated with **Consumer Insights team** to analyze Amazon headphone reviews from Beats and competitors using Python, TextBlob, Pandas, Plotly, and NumPy for **sentiment and feature extraction**, informing product feature development and marketing strategy
- Engineered custom **Gemini API prompts** to analyze over **1,000 Amazon product reviews** extracting consumer sentiments over colors, most popular dates for **product release, feature mentions and pain points**

Projects

NBA Player Stats App

- Built a **Streamlit** web app that connects to the official **"nba_api"** to pull real-time player stats, game logs, and league leaderboards. The app auto detects when current-season data isn't yet available and gracefully falls back to the previous season, it also follows **WCAG** principles to provide an accessible smooth user-experience
- Implemented **Scikit-learn Random Forest Regressor** to forecast a player's next-game performance based on **80% historical game data** from all seasons. The model dynamically trains and evaluates itself using **Mean Squared Error (MSE)** to highlight accuracy from the other **20% game data**
- Designed predictive visual, interactive insights powered by **Plotly, Pandas, PIL and NumPy**. Features a responsive layout with player headshots, team information, and statistical summaries. Added advanced visualizations like shot charts, spider plots and performance trend graphs for **comprehensive, media-grade analytics**