Brady McAtee

bradymcatee12@gmail.com | bradymcatee.com | linkedin.com/in/brady-mcatee | github.com/bradymcatee

Education

Portland State University, BS in Computer Science, Minor in Physics — Sept 2023 – March 2026

- GPA: 3.9/4.0
- **Coursework:** Data Structures and Algorithms, Operating Systems, Software Engineering, Computer Graphics

Experience

Software Engineer, Code PDX (Volunteer) – Portland, OR — Dec 2024 – Present

- Developing geospatial data management system for openTwin project using PostgreSQL/PostGIS, enabling real-time urban infrastructure monitoring
- Implementing RESTful APIs and database architecture for efficient handling of time-series and location-based data
- Collaborating with cross-functional team using Docker containers and modern Git workflow

Projects

recipeBox — github.com/bradymcatee/recipeBox

- Engineered full-stack web app with React, Node.js, and PostgreSQL, enabling recipe management for restaurants
- Designed and implemented RESTful API with JWT authentication, supporting role-based access control and data isolation between restaurants
- Implemented comprehensive unit tests using Jest, achieving >80% test coverage for frontend & backend components
- Optimized frontend performance with React hooks and state management for recipe storage and user administration
- Deployed production build on an AWS EC2 instance, allowing restaurant employees to access the app

OpenBudget — github.com/bradymcatee/OpenBudget

- Developed full-stack financial management application using Python (FastAPI), React, and PostgreSQL
- Implemented core features including user authentication, database modeling, and API development
- Applied software engineering best practices: version control, testing, and containerization

C++ Ray Tracing Engine — github.com/bradymcatee/RayTracer

- Developed C++ ray tracing engine featuring physically-based rendering, multi-threaded computation, and scene optimization
- Implemented advanced graphics algorithms including ray-object intersection, reflection mapping, and photorealistic material rendering
- Enhanced performance through spatial partitioning and parallel processing, resulting in 60% faster render times

Plant Disease Classifier — huggingface.co/spaces/diseases

- Built and trained deep learning model using PyTorch and ResNet architecture, processing 50,000+ plant images with 99% accuracy
- Implemented data augmentation techniques to improve model generalization across diverse plant species
- Optimized model deployment on Hugging Face Spaces, enabling real-time disease diagnosis for agricultural researchers

Technologies

Languages: C++, C, Python, Java, JavaScript, TypeScript, SQL

Technologies: React, Node.js, Express, FastAPI, PostgreSQL, Git, PyTorch, Jest, PyTest, Linux, Vim