# Christopher Guard Hunt | Software Developer

[**huntcg@outlook.com**](mailto:huntcg@outlook.com) **| 828-707-5333**

Highly effective problem solver who is passionately dedicated to continuous learning and creativity. Experience productively engaging with exceptionally complex business problems, designing and implementing well-tested software solutions, and thoroughly documenting processes and code.

## Education

**Berea College, Berea, KY** – Bachelor of Arts in Computer and Information Science, May 2017

## Technology

* Languages: Python, SQL, HTML, PHP, IBM RPG
* Versioning: Git, Github, Bitbucket
* Frameworks: Flask, Laravel
* Python Libraries: Requests, SQLAlchemy, BeautifulSoup, Alembic, Marshmallow, Click, Pandas, PyODBC
* Databases: MySQL, Elasticsearch, IBM DB2, SQL Stored Procedures, ODBC
* Systems/Platforms: Ubuntu, Nginx, Werkzeug, AWS (Cognito, SNS), Docker, IBM iSeries

## Technical Skills

* RESTful API development, documentation, and implementation
* Containerization: docker, docker-compose, docker-swarm
* Automated test writing and maintenance: unit tests, fixtures, mocking, dummy data
* Relational database design and management: normalization, entity/attribute/relationship differentiation, foreign-keys, indexing, views, handling migrations
* Document-store database design and management: indexing, sharding, templating, aliasing, aging

## Experience

**Software Developer, Jackson Furniture Ind. (July 2019 – Current)**

* Furthered application improvements and bug-handling in existing python applications
* Became competent in IBM RPG to maintain and improve accounting tools

**Junior Python Developer, Jackson Furniture Ind. (August 2017 – July 2019)**

* Developed several RESTful microservices: written in Python with the Flask framework, tested with the Pytest library, deployed with docker and docker swarm, used a combination of Redis and python-Dogpile for caching to achieve fast response times
* Created a cross-application authentication and authorization service built on-top of AWS Cognito, using the JWT standard to centralize user account data, manage a freeform role-based authorization model, and allow for single-sign-on and multi-factor authentication
* Designed and implemented a full Elastic Stack cluster for application traffic and error logging across multiple web applications, in addition to full-text search on product attributes for our consumer-facing website
* Re-wrote the logging implementation for a large PHP web application using the monolog library inside of the Laravel framework, generating JSON documents with proper structure, fields, and nesting to be ingested by the Elastic Stack
* Wrote a small full-stack web application on top of Elasticsearch, which linked application error logs with user traffic logs by using unique session request keys, providing time-based quires by user, and offering insights from front to back-end application requests for handling user-reported errors