Leon-Hunter-Public-Portfolio

View the Project on GitHub at https://github.com/

Click here to download the most recent `.pdf`

Senior Cloud Native Application Developer

Junior Cloud DevOps Engineer

About

Contact Information

- **Phone**: `(302) 312-4489`
- **Email**: `xleonhunter@gmail.com`
- **Website:** `https://git-leon.github.io/Leon-Hunter-Public-Portfolio`
- **LinkedIn:** `https://www.linkedin.com/in/leon-hunter-3230408b`
- **Github Organization Ownership**
  - `https://github.com/curriculeon`
  - `https://github.com/git-leon`

Education

- Delaware State University (2011 - 2015)
- Bachelor of Science (BS)
- Computer Science Major, Mathematics Minor

Summary

Experienced T-shaped Engineer and Technical Instructor with a demonstrated history of managing and up-skilling teams of Junior engineers. Highly skilled in Cloud Native Application Development using Java8, Spring, Mockito, and Angular cumulating more than 1800 total Git contributions in the past year. Knowledgeable in Cloud Dev Ops using unittest, JUnit5, Selenium, Docker, Kubernetes, Jenkins, and AWS. Strong information technology professional with a Bachelor of Science (BS) focused in Computer Science, Mathematics Minor from Delaware State University.

Personal Projects

Owner and Creator

Curriculeon: The Online Coding Curriculum

- Curriculeon is an online curriculum built in RevealJs, Jekyl, and Github Pages, which offers free open-source learning content for the following topics:
  - Java Web Development

    - Command Line Scripting
    - GitCli
    - HTML/CSS/JS
    - Python Scripting
    - Cloud Computing & Cloud Infrastructure
    - Development Operations - (DevOps)

- The following is a list in ascending order of delivery date of some of the relevant processes that were covered and applications that were built during a 14-week program in a single lecture (**90 minutes or less**). Demonstrations relied heavily on git, TDD and Cloud Native practices; enforcing version-control, portability, and testability across each of the developer's environments. It should be noted that each application, even the creation of the Github repository, was built in front of a live audience of 25 to 30 developers, while polling for and answering questions.
    - A simple JUnit Testing application built with Maven using JUnit5 testing paradigms and conventions for testing input and output of algorithms;
        - The intent of building this application for the developers is to demonstrate how to implement proper TDD practices. It is critical that this is presented early to the developer to ensure that each demonstration that follows can be built and then tested appropriately.

    - A crude database-access application built with Maven using JUnit5 and JDBC to test database connectivity, querying capabilities, and database insertion;
        - The intent of building this application for the developers is to demonstrate how to leverage Java's JDBC API to connect to delete, create, and connect to a new connection. The demonstration included how to switch Driver implementation by modifying the `pom.xml` to specify the JDBC driver.

    - A scalable database-access application built with Maven using JUnit5 and JDBC to test database connectivity querying capabilities and database insertion
        - The purpose of demonstrating how to build this application is to assist the developers with establishing how to build an application that leverages strictly JDBC and scales easily by adding more POJO-oriented logic.
        - This demonsstration will prepare developers for proper application-design and scalability by abiding by SOLID principles and introducing GoF Design Patterns (particularly factories, builders, and singletons).
        - In the days that follow, the developers witnesses how proper design can scale flexibly. The decision to implement this POJO-oriented logic is to later enforce an entity-driven-design implementation using JPA and again later with a Spring Boot implementation.

    - A scalable-database access application built with Maven using JUnit5 and JDBC and JPA.
        - The developers witnesses how the aforementioned application is easily transformed into a JPA project by introducing `persistence.xml` in the `main/resources/META-INF` directory and including a JPA flavor in the `pom.xml`.

    - A Spring application which exposes basic CRUD operations of a controller managing an entity with implicit @Basic field-types
        - By leveraging the H2-console, the developers witness the entire development lifecycle of a Spring Boot application beginning with empty tables. As an Entity is defined, the developers immediately view the change in the console.
        - The developers benefits from viewing this demonstration by witnessing how Beans, EntityManager, IoC container, and Dependency Injection work in the context of Spring
        - The application is tested and interacted with via Postman

- A Spring application with automated tests for Controllers, Services, and Repositories by using BDDMockito
  - Using Junit & SpringRunner, MockMvc, MockBean, BDDMockito, and Jackson databind, the developers learns how to create automated tests to check the correctness of the web server. This step is critical in the developers' advancement toward automation. This ensures that each of the applications that follow can be provisioned with automated tested.

- A Spring application which exposes CRUD controllers with multi-dependent services and entities with relational cardinality
  - The developers establishes how to make use of an Entity with a cardinal relation to another Entity (1-1, 1-M, M-M, M-1). Additionally, they view how to inject multiple dependencies from the IoC container into the scope of a dependee.

- A Spring Web Server and JSP Web Application which renders a webpage with content injected from the web server
  - This application allows the developers to see how to communicate throughout their entire application-stack: H2 in-memory database, Spring WebServer, JSP Web Application.

- A Spring Web Server and JSP Web Application which renders a login page to create and manage User sessions within the system
  - Using Spring security and the aforementioned technologies, the developers learns how to create an application which enables sign-up, and login features. Additionally, the application manages User sessions for multiple connections and instances.

- A Spring Web Server and ReactJS Web Application which allows a client to add and view ReactJS Web application which renders a list of customers fetched from a Spring Web Server
  - The purpose of building this application is to demonstrate how microserver architecture allows us to easily swap implementations of a layer of the stack

- A Spring Web Server and Angular6 Web Application (Spring Web Server and Angular6 Web Application) which allows a client to add and view a list of customers fetched from a Spring Web Server
  - The purpose of building this application is to further expound how microserver architecture allows us to easily swap implementations of a layer of the stack

- A continuous integration system (Git, Jenkins, Web Server, Web Application) which fetches a remote repository, builds the application, and runs tests, and reports the results to Jenkins
  - The intent of building this system is to demonstrate how to create an automated pipeline for continuous integration and continuous delivery

- A demonstration of containerizing Spring jars
  - This demonstration allows developers to learn how to containerize applications using Docker with the intent to deploy the application in the most portable way possible.

- A demonstration of deploying Spring jars to AWS Elastic Beanstalk
  - The intent of this demonstration was to provide the developers with industry standard practice for deploying and managing applications on the cloud.

Owner and Creator

Leonium.jar: A Selenium and ExtentReport Wrapper Library

- Leonium is an uber jar deployed on packagecloud.io that is used to automate Selenium build configuration and script development.
- The library includes an embedded `.jar` used for dynamically downloading driver binaries
    - This allows this framework to run `GeckoDriver`, `ChromeDriver`, `PhantomDriver`, `HtmlUnitDriver` without downloading any external dependencies

- The library also includes mechanisms for
    - enabling logging for each web-interaction
    - enabling reporting using ExtentReports for aggregating results
    - helper classes with built-in intelligent-waits

Owner and Creator

OpenCVL.jar: An OpenCV Wrapper With Haar Cascade Machine Learning

- OpenCVL is an uber jar deployed on packagecloud.io that is used to ease implementation of Object Detection Algorithm by leveraging OpenCV and HaarCascadeClassifiers.
- The library includes mechanisms for
    - a factory class for dynamically fetching haar cascade classifiers
    - face detection interface
    - default face detection implementation with CascadeClassifier dependency injection support
    - face detection builder class
    - face detection factory class

Owner and Creator

JFoot.jar: A Greenfoot Wrapper Library

- Jfoot is an uber jar deployed on packagecloud.io that is used to make Greenfoot applications with Maven.
- The library includes mechanisms for
    - converting greenfoot applications to maven applications
    - helper interfaces for building games
    - small gravity physics engine

Owner and Creator

ExceptionalFunctionalInterfaces.jar: A decoration of Java8 FunctionalInterfaces

- Exceptional Functional Interfaces is `.jar` deployed on packagecloud.io that is used to express code that explicitly throws an `Exception`, without ever explicitly handling it by defering the `try`/`catch` to the respective `FunctionalInterface`'s static `tryInvoke` method.
- With the advent of Java 8 Lambdas and Method References, functional programming in java has become increasingly popular. Unfortunately, Java 8 does not account for creating lambda expressions which may throw an `Exception`. This shortcoming has caused a great deal of frustration when developing an exceptional-codebase which should be expressed functionally rather than with object orientation. This library aims to help rememdy the deficiencies of exceptional expressions in Java 8 functional programming.

Experience:

Remote Lead Technical Instructor

Per Scholas: Cloud DevOps Engineering Course

November 2020 – February 2021; Boston, MA.

- Remotely leading a group of 30 aspiring engineers through a Cloud DevOps environment.
- teaching the following topics through live-demonstrations to an audience of 25 in daily Zoom meetings:
    - GitOps, Git, Shell scripting, SQL / DBMS, TDD, Python Scripting
    - Cloud Infrastructure, Cloud Security
    - CI / CD, Nginx, Build Management
    - Deployment and Jenkins Pipeline
    - Docker and Kubernetes
    - AWS Terraform, AWS ECS, AWS EKS, AWS Lightsail, AWS EC2, AWS Cloud9

Remote Lead Technical Instructor

Talent Path: Full Stack Java Engineering Course

November 2020 – December 2020

- Using Git, Zoom, Slack to manage a group of developers through the development of Java Web-applications.
- Leveraging Curriculeon curriculum lectures and curriculum exercises to launch learners into an immediately immersive development environment to enforce a deploy-on-day-1 culture.
- Learners receive morning lectures from the curriculum, then exercises from the curriculum to complement the lecture.
- The following is a list in ascending order of delivery date of some of the relevant content presented to learners.
    - A Java assessment built with Maven using JUnit5 testing paradigms and conventions for testing input and output of algorithms;
        - The intent of building this application for the developers is to demonstrate how to implement proper TDD practices. It is critical that this is presented early to the developer to ensure that each demonstration that follows can be built and then tested appropriately. This also gives the instructor the opportunity to gauge the classes aptitude. With this group, it was the case that they had strong understanding of the Spring framework, but poor understanding of Object Orientation, Design Principles, and Design Patterns. After making this discovery, the focus was to upskill the learners in the aforementioned areas.

    - A Spring application which exposes basic CRUD operations of a controller managing an entity with implicit @Basic field-types
        - By leveraging the H2-console, the developers witness the entire development lifecycle of a Spring Boot application beginning with empty tables. As an Entity is defined, the developers immediately view the change in the console.
        - The developers benefits from viewing this demonstration by witnessing how Beans, EntityManager, IoC container, and Dependency Injection work in the context of Spring
        - The application is tested and interacted with via Postman

    - A system of classes enforcing several singleton design-pattern implementations, repository design pattern implementation, and decorator design pattern implementation.

        - The purpose of building this application was to demonstrate the purpose of scalable design decisions and how cutting corners in design can create technical debt and violations of DRY.

    - *More content will be delivered as the course continues*

Remote Curriculum Developer

Per Scholas: Cloud DevOps Engineering

September 2020 – November 2020.

- As a remote curriculum developer, responsibilities included creating content for curriculums which extensively cover the following topics and more over a twelve to fourteen week period:
    - Cloud Security
    - Cloud DevOps
    - Cloud DevSecOps
    - Java8, Spring Security, Spring MVC, Spring Boot, JPA, JDBC, JSPs, Mockito, Git

Remote Lead Technical Instructor

Per Scholas: Full Stack Java Engineering Course

June 2020 – September 2020; Charlotte, NC.

- Created more than 65 github repositories, along with branches and configured branching rules, for trainees to leverage to build applications, cumulating more than 550 Git contributions within the organization during the 14 week program.
- Managed the group of developers through the development of more than 20 functioning full stack java applications of developers by utilizing Git and Cloud Native Application Development practices. Application stacks used some variance of the technologies listed below.
    - Java8, Spring Security, Spring MVC, Spring Boot, Mockito, JSP, ReactJS, Angular
    - AWS S3, AWS EC2, AWS Cloud9, Heroku, Netlify

Remote Curriculum Developer

Per Scholas

March 2020 – June 2020; Wilmington, DE.

- As a remote curriculum developer, responsibilities included creating content for curriculums which extensively cover the following topics and more over a twelve to fourteen week period:
    - A+ CompTIA
    - Network Security
    - ReactJS, Redux, Node
    - Java8, Spring Security, Spring MVC, Spring Boot, JPA, JDBC, JSPs, Mockito, Git

Lead Software Engineer

Scriptcasters.com: OverseasBasketballConnection.com

January 2020 – May 2020; Wilmington, DE.

- Cofounder of Scriptcasters.com, a network of specialized professionals that strive to create innovative technical solutions.
- Hired and lead team of developers to create and maintain a web application and administrative portal for a client seeking a tech solution
- Lead the creation of an application for a client which can connect professional basketball players with overseas teams seeking players, overseasbasketballconnection.com.
- Working alongside one other developer, Frankie Rodriguez, we together created a very successful, popular, heavily used 3-tier platform which
    - allows agents to post job opportunities for players to apply to
    - allows players create a profile and apply for opportunity postings
    - admin portal, which gives client control administrative control over the two aforementioned tiers.

- Including integration testing, deployment, and built configuration, I built the following aspects of the application in their entirety:
    - Java8, Spring MVC, Spring Security, AWS S3, Heroku, Selenium

- The following APIs were leverages for additional features:
    - PrintfulCountry API
    - MailChimp API

Lead Technical Instructor

Per Scholas: Full Stack Java Engineering Course

November 2019 – February 2020; Philadelphia, PA.

- Lead Technical Instructor at Per Scholas's Philadelphia pilot of Full Stack Java Engineering bootcamp.
- As lead technical instructor, responsibilities included covering course material via lecture as well as building and testing applications live in front of an audience of 25 developers.

Lead Curriculum Designer

CodeDifferently

July 2019 – November 2019; Wilmington, DE.

- As the Lead Curriculum Developer, responsibilities include technically instructing learners, designing exercises and solutions, as well as managing 3-person development team.
- Created lesson plans, lecture sequences, and architected teaching material via abstractions of reusable content

Lead Curriculum Developer

Zip Code Wilmington: Full Stack Java Engineering Course

December 2018 – May 2019; Wilmington, DE.

- Lead content developer for java programming boot camp.
- Created solutions, lectures, and curriculum for training program.
- Taught all core concepts of java and advanced topics including JDBC, JPA, Spring Boot, and design patterns
- Assisted in the education and job-placement of over 200 junior full-stack application developers

Contractor / Technical Instructor

JPMorgan Chase & Co.

October 2018 – November 2018; Glasgow, United Kingdom

- Contracted through ZipCodeWilmington, I served as a technical Java instructor teaching web-development to JP Morgan's incoming, and transitioning developers located in Glasgow, Scotland.
- The course was an intense 12 week training program mocking a real-world developer environment by enforcing collaboration strictly through Github, and enforcing 12 factor configuration methodologies such as leveraging Maven for dependency management and environment variables for application configuration. The course was taught using a stack of Maven, SpringBoot, MySQL, Angular and Heroku

Contractor; Lead Technical Instructor

JPMorgan Chase & Co.

June 2018 – September 2018; Wilmington, DE.

- Contracted through ZipCodeWilmington, I served as the technical Lead Java instructor teaching web-development to JP Morgan's incoming, and transitioning developers.
- The course was an intense 12 week training program mocking a real-world developer environment by enforcing collaboration strictly through Github, and enforcing 12 factor configuration methodologies such as leveraging Maven for dependency management and environment variables for application configuration. The course was taught using a stack of Maven, SpringBoot, MySQL, Angular and Heroku.

Lead Technical Instructor

Zip Code Wilmington

April 2018 – December 2018; Wilmington, DE.

- Lead content developer for java programming boot camp.
- Created solutions, lectures, and curriculum for training program.
- Taught all core concepts of java and advanced topics including JDBC, JPA, Spring Boot, and design patterns
- Assisted in the education and job-placement of over 200 junior full-stack application developers

Contractor; Technical Instructor

JPMorgan Chase & Co.

June 2017 – September 2017; Wilmington, DE.

- Contracted through ZipCodeWilmington, I served as a technical Java instructor teaching web-development to JP Morgan's incoming, and transitioning developers.
- The course was an intense 12 week training program mocking a real-world developer environment by enforcing collaboration strictly through Github, and enforcing 12 factor configuration methodologies such as leveraging Maven for dependency management and environment variables for application configuration. The course was taught using a stack of Maven, SpringBoot, MySQL, Angular and Heroku.

Technical Instructor

Zip Code Wilmington

April 2017 – April 2018; Wilmington, DE.

- Taught all core concepts of web design in the context of Java.
- Covered advanced topics including JDBC, JPA, Reflections, Collections.
- Lectured on design patterns in the context of advanced frameworks such as Spring Boot and Angular.
- Assisted in the education and job-placement of over 200 junior full-stack application developers

Test Automation Engineer

Marlette Funding

January 2016 – April 2017; Wilmington, DE.

- As a test automation engineer, I was responsible for the implementing a WebCrawling framework for UI automation scripts and error reporting.
- The application was implemented using Selenium, Selenium Grid, SauceLabs, TestNG, and Extent Reporting.

Undergraduate Researcher

Optical Science Center for Applied Research (OSCAR)

August 2015 – December 2015; Dover, DE.

- The objective of this project was to develop a genetic algorithm to control a spatial light modulator (SLM) used for shaping the profile of a laser beam.
- Implemented evolutionary algorithm for adaptive optics using MatLAB
- Interfaced MatLAB with LabVIEW software.

Operations Leadership Intern

Discover Financial

June 2015 – August 2015; New Castle, DE.

- Implemented BEST QCAS Access Database.
- Manual web tester and bug reporter.
- Implemented Microsoft Word and Excel macros using VBA to partially automate test reporting.

Programming Tutor

Delaware State University

January 2015 – May 2015; Dover, DE.

- Aided learners with completing projects developed in python & java.
- Assisted with understanding of programming logic.

Undergraduate Researcher

National Science Foundation (NSF)

May 2013 – May 2014; Dover, DE.

- The objective of this research was to develop a system that would aid in the prevention of catastrophic medical events through persistent intelligent monitoring and early-warning alerting.
- Created interactive electronics using Arduino open-source prototyping platform.
- Coded in C++ to manipulate Arduino microcontroller and eHealth sensor shield.
- Presented research for 2014 Science and Math Investigative Learning Experiences Program (SMILE) Undergraduate Summer Research Symposium.
- Presented at 2014 ERN (Emerging Researchers National) conference.

Programming Tutor

Delaware State University

January 2013 – May 2013; Dover, DE.

- Aided learners with completing programs developed in java.
- Assisted learners with conceptualizing and understanding programming logic.