MODULE 7

7-1: Final Project

CS499 Computer Science Capstone

Zhraa Al-Bayaa

This paper covers the final project to represent the great journey taken throughout this course CS499 Computer Science Capstone, its assignments, modules, development of ePortfolio, and final project along with Github website using its pages API. This paper will cover the complete reflection of my journey, achievements, and accomplishments evolving ePortfolio and enhancement of three artifacts covering Software Design and Development, Databases, and Algorithms and Data Structures. The ePortfolio will display and showcase skills learned and evolved during this course, the ability to practically connect through networking, and present myself as compatible with the lucrative Software Development Job Market.

There are competitive advantages projected by creating a professional ePortfolio to showcase and present my unique skillset, abilities, and competence to attract potential companies and employers showing my compatibility for the Job Market. The creation of an ePortfolio allows me to highlight my successive growth, the evolution of my learning, and the enhancement of my skillset over time from course to course and subject to subject. It covers my accomplishments, interests, inclination, leadership, teamwork, problem-solving techniques, and research work. It demonstrates how I accomplished my skillset, assignments, enhanced artifacts, and projects over time during this course. It provides a rich and prolific resource to present the details necessary to showcase my compatibility to potential employers and companies seeking to hire a competent workforce. The course CS499 Computer Science Capstone helped me learn how to evolve ePortfolio by understanding how to improve current projects selected as artifacts at the beginning of the course. During the modules of this course, I learned how to improve these artifacts by identifying the problems and vulnerabilities in the current projects.

This course helped me increase my understanding and skills to understand problem-solving, identifying problems and shortcomings in current applications, and ways to improve existing projects. Throughout this course, I carried out analysis, problem-solving, software development, artifact enhancement, and technical writing while evolving my ePortfolio to present my achievements and accomplishments throughout this course and list the projects from previous courses. All the courses I have studied during this Computer Science program have helped enhance my skillset, improve my communication, problem-solving skills, requirement gathering, analysis and design, and application development using high-level programming languages and not the least reverse engineering and working with embedded programming application. All these achievements, skills, projects, and knowledge are necessary to secure a good and lucrative position in some job related to software application development as a software engineer.

Quality assurance is a vital part of the software development process as it is a systematic approach and process to assess the compatibility of an application to meet its functional and non-functional requirements. Quality assurance conforms to a set of requirements and establishes the reliability of the project. Usually, it is performed after the testing phases of the application development. Code review is a systematic and methodological assessment of the application code to identify and recognize errors, bugs, and logical problems in an existing code base. Through Code Review we can easily catch bugs and errors in an existing code and fix them before the application goes for final testing and quality assurance. At the early stage, it is cheaper to debug, fix, maintain, and improve the code as compared to carrying out this process after the deployment of the application. Code Revier offers many benefits to application development, as it helps improve the code and enhance the functionality of the application. As it is reasonably sized in terms of the amount of code being altered to enhance the application, it takes less time and provides better results. During this course, I have selected three artifacts for Software Design and Engineering, Algorithms and Data Structures, and Database. I have enhanced these artifacts diligently by identifying problems, proposing new improvements, code arrangements, and algorithm improvements in terms of performance, efficiency, and speed, boundary testing, unit testing, and increased search by optimizing indexes, while developing the code review video for all the three artifacts I arranged and developed my transcript to represent and cover all the aspects of this artifact enhancement. Discussed the importance of Code Review, located bugs and errors in the code, and proposed improvements and enhancements, screen captured code editing with live commentary recorded in the video, and explained every process and step taken to perform the Code Review of all these three artifacts. I walked through the audience line by line, step by step for each of these Code reviews performed on these artifacts. During this code review, I enhanced my skills to present my coding skills, problem-solving, communication, and oral and visual presentation of the code and applications to attract a diverse audience in the field of software application development. The video presentation of these code reviews of all three artifacts is not only appealing but also technically sound offering competitive knowledge to the appropriate audience in the field of software application development.

First Artifact: The artifact selected for this software design and engineering category is related to a Project completed during a course taken at (Southern New Hampshire University) CS320. This project was related to creating a Java-based Contact class along with a repository class to provide crude operations to manage the in-memory contact repository and service. The project also involved creating unit tests for both these classes Contact.java and ContactService.java. The application intends to manage contacts and uses Java HashMap for an efficient in-memory data repository and provides robust unit tests to validate the data input and intended functionality to improve the quality assurance of the artifact. To create this application, java’s JDK version 18 was used in combination with a famous Eclipse IDE. The Junit Test version

5 is used to test the boundary conditions, functionality, and data input validations. The Entity classes is used following the best coding standards using data encapsulation, constants, and robust functions to create and use Contact objects. The repository class used HashMap which is one of Java’s fastest collection frameworks with efficient insertion, retrieval, and deletion procedure runtimes.

The creation and enhancement of this application showcase the skills I learned during this CS320 course to learn about software development and quality assurance by creating automated unit tests to test and validate the functionality and correctness of the application. Dynamic application testing offers various advantages to achieve the robustness, correctness, performance, efficiency, and improved quality assurance of the application being developed. Unit testing and quality assurance bring many competitive advantages to the software applications that undergo rigorous testing such as saving maintenance costs, consistent performance as expected, enhanced brand reputation, improved efficiency and operational performance, reduced bugs, and achieving compliance with the regulatory bodies. Testing functionality and boundary conditions through Unit testing ensure that an application will always produce the intended results for a given dataset. During the development and enhancement of this artifact, I followed the best coding standards and programming practices while closely adhering to the software design and development phases. I used an efficient data structure to manage the contact and wrapped it into a repository class classed ContactService. Java’s HashMap is one of its fastest data structures concerning its efficiency it has time and space complexity of O (N) and operations O (1) times at average and O(N) in the worst-case scenario. These runtimes prove that the chosen data structure is fast enough to service the requested CRUD operating for this in-memory contact repository.

Second Artifact: This artifact has been selected for the Algorithms and Data Structure Category from CS300 Data Structures and Algorithms: Analysis and Design. The application demonstrates reading datasets from a CSV File related to Bids loaded the data set into a Vector. The Vector is then sorted using two different sorting algorithms Selection Sort and Quick Sort. This application is developed using C++ programming language and Visual Studio Code is used as an IDE with g++ GNU Compiler to compile the source code on Windows operating system. This application and artifact involve File IO, Data Validation, Structs, Methods, Sorting Algorithms, and User Interaction via a Text-based menu. The artifact was developed using the best software development principles, secure coding, and best programming methods to address the intended application goals and objectives. The secure coding methodologies were adopted to reduce C++ coding vulnerabilities and reduce the risks of any malicious usage and invalid memory access by adopting measures to eliminate any memory overlapping and illegal access by limiting the vector size boundaries and indices. The data set has been loaded from an external file into the dynamic vector data structure that accommodates the data as long as memory is available in the operating system and avoids any memory exploitation. The application offers a Text-based menu to interact with the application and the user enters a menu option to carry out different functions offered by the application. This application uses a modular approach to address the application functionality that is extensible and easily adopted for future enhancements.

The enhancement of this artifact's industry-standard C++ coding rules and practices, minimized security vulnerabilities, memory violations, and buffer overflow. Coding is written with standard C++ conventions, naming conventions, indentation, self-explanatory variable names, and internal documentation with inline and block-level comments. The code is easier to read and comprehend, extensible, and maintainable because it is well structured and formatted to industry-level C++ standards. The code is tested manually to fix any logical error. The input errors are handled to avoid any program failure or crashes during the application execution. The code is compatible to run on Windows, Mac, and Linux using the latest g++ GNU Compiler with any IDE or simple text editor. The application can handle any size of data or any number of Bid records loaded from a CSV File as long as the file is properly formatted.

Third Artifact: The selected Artifact is chosen from a project completed for the Course CS340 Client and Server Application Development. The application involves creating a document-based MongoDB NoSQL Database from a given Dataset comprising Animal data for a given Animal Shelter. The CS340 Client and Server Application Development course teaches how to plan, design, and develop client/server applications. The selected artifact is based on the creation of a NoSQL database from the given dataset. The suggested enhancement to this database application is related to creating Simple and Complex indexes to enhance the database search routines. MongoDB database systems help optimize and improve the execution of data selection or projection queries by incorporating indexes. Indexes support the efficient execution of the select queries to retrieve and filter the query-based data for its efficient and improved performance. Without indexes, the queries usually become sluggish, especially in the case of larger datasets. Incorporating indexes not only improves the database performance but also decreases the runtime of the query execution significantly. In the absence of Indexes, MongoDB scans every document present within the database which is like a brute force operation taking a lot of time and sluggish in preparing an output result. By introducing indexes, MongoDB limits the document related to that particular index and reduces the time complexity improving the runtime and database performance that helps develop robust and efficient systems that may produce results in real time without lagging the system. The indexes in MongoDB database System work by storing the specific field value or set of field values by the order of that particular simple or complex index. These ordered indexes match the query parameters and produce quality results filtered quickly as they are comprised of smaller datasets based on the indexes instead of every document in the database resulting in faster query output. The following screenshot shows the database created with 10k records/documents each representing data on a certain animal within the AnimalShelter and 4 indexes were created to enhance the query execution and search functionality within the database.

Based on these artifacts, my ePortfolio showcases my abilities and skillset based on the latest technology and programming basis. The projects I completed have a strong foundation in software development and industry-specific goals and objectives covering Java Programming language with Junit testing, MongoDB NoSQL database, and C++ Programming languages with a strong association with algorithm development and implementation. Worked with C++ and Java core programming languages with extensive exposure to best programming practices and conforms to the language standards with the least vulnerability and secure coding practices. Self-assessment depicts that my skillset and accomplishments are compatible with latest Computer technology and trends. I am ready to work in a diverse workspace of software application development. I am ready to work in a diverse team with different skills and collaborate in the development of industry standard and enterprise level applications using Java Programming, Junit Testing, C++ Programming, Relational and No-SQL Databases, Algorithms and Data structures, Software design and development, Scrum Framework, and agility, High level and low-level languages, Android programming using Java app development, Full stack development using React and Angular. This skillset is prolific and comprehensive to introduce myself in the competitive job market of software application development.