|  |  |
| --- | --- |
|  | |
| PROG7312 POE Part 3 | |
| Student Number: | ST10144348 |
| Programme Code: | BCA3 |
| Module Lecturer: | Reece Wanvig |
| Module Code: | PROG7312 |
| Date of Submission: | 18-11-2024 |

|  |
| --- |
| I hereby declare that I did not plagiarise the content of this assignment and that this is my own work.  Assignment submitted via SafeAssign: (Tick the Box) |

Table of Contents

[1 Implementation Report 1](#_Toc182847901)

[2 Project Completion Report 2](#_Toc182847902)

[3 Key Learnings 3](#_Toc182847903)

[4 Technology Recommendations 4](#_Toc182847904)

[5 Reference List 5](#_Toc182847905)

# Implementation Report

**Data Structures Used**

**Binary Search Tree (BST):**

* Role: Efficiently organizes and retrieves service requests based on unique IDs.
* Contribution: Enables fast lookup and traversal operations, ensuring the application's scalability.
* Example: Service requests are stored in a BST with IDs as keys, allowing for quick retrieval and sorted order display.

(Geeksforgeeks, 2024)

**Graphs:**

* Role: Represents relationships between service requests (e.g., dependencies or stages).
* Contribution: Enhances visualization and management of interdependent service requests.
* Example: Nodes represent service requests, and edges indicate dependencies between tasks.

(Geeksforgeeks, 2024)

**Heap:**

* Role: Prioritizes service requests based on urgency.
* Contribution: Ensures high-priority tasks are highlighted or processed first.
* Example: A min-heap orders requests by their urgency levels.

(Geeksforgeeks, 2024)

# Project Completion Report

**Summary of the Project**

The project aimed to create a feature Service Request Status page integrated into the Municipal Services Web Application. This feature provides users with detailed insights into the progress of their service requests.

**Challenges Faced**

**Efficient Data Management:**

* Challenge: Implementing advanced data structures to manage large volumes of requests efficiently.
* Solution: Used BSTs for request organization and graphs for dependency tracking.

**User-Friendly Interface:**

* Challenge: Designing a UI that aligns with the application's overall look and feel.
* Solution: Incorporated Bootstrap and custom CSS for a clean, responsive design.

**Integration with Backend:**

* Challenge: Ensuring seamless data flow between the local storage of data and the frontend.
* Solution: Used trees and graphs to store and retrieve data.

# Key Learnings

**Advanced Data Structures:**

* Gained a greater understanding of implementing and using BSTs, graphs, and heaps in web applications.

**Problem-Solving Approaches:**

* Enhanced skills in debugging, optimizing performance, and ensuring compatibility across system components within the web application.

**Programming Techniques:**

* Improved proficiency in Razor Pages, C#, and integrating frontend frameworks with backend logic.

# Technology Recommendations

**ElasticSearch:**

* Purpose: Enhances search capabilities for service requests.
* Benefits: Provides full-text search, filters, and sorting for large datasets.
* Justification: Improves the user experience by enabling rapid and precise query results.

(Elastic, 2024)

**Docker:**

* Purpose: Containerizes the application for consistent development, testing, and deployment environments.
* Benefits: Simplifies application deployment, ensures scalability, and reduces compatibility issues across different systems.
* Justification: Docker enhances reliability and streamlines DevOps workflows, making it easier to deploy updates and maintain the application.

(Nnanna, 2024)

**Twilio:**

* Purpose: Provides SMS and email notification services to update users on the status of their service requests.
* Benefits: Ensures timely communication with users, keeping them informed about progress and any updates.
* Justification: Easy integration with web applications and improves user satisfaction by offering real-time updates on service requests.

(Twilio, 2024)

# Reference List

Geeksforgeeks (2015). Binary Search Tree - GeeksforGeeks. [online] GeeksforGeeks. Available at: <https://www.geeksforgeeks.org/binary-search-tree-data-structure/>.

GeeksforGeeks (2019). Difference between graph and tree. [online] GeeksforGeeks. Available at: <https://www.geeksforgeeks.org/difference-between-graph-and-tree/>.

GeeksforGeeks. (2015). Why is Binary Heap Preferred over BST for Priority Queue? [online] Available at: <https://www.geeksforgeeks.org/why-is-binary-heap-preferred-over-bst-for-priority-queue/>.

Nnanna, S. (2024). Containerization and Deployment of a Java Application using Docker and Kubernetes. [online] Medium. Available at: https://medium.com/@samuelnnanna71/containerization-and-deployment-of-a-java-application-using-docker-and-kubernetes-0286d948b3ee [Accessed 18 Nov. 2024].

Twilio. (2023). Twilio Messaging Engagement Suite. [online] Available at: https://www.twilio.com/en-us/use-cases/alerts-and-notifications [Accessed 18 Nov. 2024].

www.elastic.co. (n.d.). Search API | Elasticsearch Reference [7.11] | Elastic. [online] Available at: <https://www.elastic.co/guide/en/elasticsearch/reference/current/search-search.html>.