# Technical Assessment Specification

## Overview

This assessment is designed to evaluate your skills in C#, SQL Server, database optimization, and your ability to discuss programming methodologies. Please read the requirements carefully, and feel free to showcase your knowledge by detailing why you made specific choices throughout the solution.

Your solution should demonstrate:

* Proficiency in SQL Server and stored procedures
* Thoughtful use of programming methodologies and design patterns
* The ability to analyze and critique your own work

## Project Requirements

You are tasked with building a C# MVC web application that performs text analysis on *Moby Dick* by Herman Melville. Your application should:

1. **Parse the Text File**: Analyze the text to determine:
   * The frequency of each word.
   * The most frequent word overall.
   * The most frequent word with exactly seven characters.
   * The highest scoring word(s) based on Scrabble scoring.
2. **Store Data**: Use SQL Server (version 16 or above) to store and query the text data. All data operations (such as inserting, updating, and querying word frequencies) must be handled by stored procedures. **Do not use raw SQL queries directly in your C# code**.
3. **Optimize with Indexes**: Create indexes on relevant database tables to improve query performance. Provide a brief analysis of your indexing strategy.
4. **Analyze Performance**:
   * Include screenshots of the Actual Execution Plan for key queries, and discuss any bottlenecks you identified.
   * Suggest at least three ways to optimize the application’s performance, specifically concerning database operations.
5. **Include ORM Considerations**:
   * Discuss any ORM choices you made (e.g., Entity Framework, Dapper, or a custom solution), and explain why you chose one approach over another.
6. **Explain Methodologies**: In your documentation, discuss any programming methodologies or principles you used, such as OOP, MVC, DRY, KISS, or YAGNI. Describe why you found these methodologies useful for this project and provide examples of how you implemented them in your code.
7. **Self-Assessment and Code Critique**:
   * Critique your own code, identifying any areas where you encountered challenges or feel the solution could be improved.
   * Discuss any trade-offs you made between maintainability, performance, and simplicity.

## Deliverables

1. **Code**: Submit your project files, including the C# MVC application and SQL Server scripts.
2. **Documentation**: Include a README file with the following:
   * Steps to set up and run the application.
   * Instructions for setting up SQL Server tables and stored procedures.
   * A summary of your ORM selection and methodology choices.
   * A list of references or resources you used to complete the assessment.
   * An approximate time log of the time taken to complete the project.
3. **Performance Analysis**:
   * Screenshots of the SQL Server Actual Execution Plan for at least two stored procedures.
   * A brief write-up on your indexing strategy and why you chose specific columns for indexing.
   * Suggestions for future performance improvements.

## Evaluation Criteria

Your submission will be evaluated based on:

* **Technical Accuracy**: Does the application meet the core functionality requirements? Are stored procedures used appropriately?
* **Performance Optimization**: Are the database operations optimized for performance? Is there clear evidence of indexing and execution plan analysis?
* **Methodology and Reasoning**: Did you explain your choices around methodologies and ORM in a thoughtful way? Are the reasons well-justified?
* **Code Quality**: Is the code clean, organized, and well-documented? Are best practices like error handling, logging, and separation of concerns followed?
* **Self-Reflection**: Does your critique highlight awareness of potential improvements and trade-offs? Have you provided constructive suggestions for future enhancements?

## Additional Notes

* Remember to document all resources you referenced, including online articles, code snippets, or documentation pages.
* Be prepared to discuss your code and reasoning in a follow-up interview.
* This assessment is meant to gauge your familiarity with the technologies, your problem-solving approach, and your attention to detail. It’s not necessary to create a perfect solution, but we are interested in understanding your thought process.

Thank you for your time and effort. Good luck!