|  |  |
| --- | --- |
| **Approach** | **Justification** |
| Using ASP.NET Core Web API in contrast to .NET Core MVC | Since the project focuses on providing a backend service for converting numbers to words, ASP.NET Core Web API is the appropriate choice. |
| Using React for frontend | Instead of using simple HTML/CSS Javascript or vue.JS or Angular, I selected React for its flexibility and performance in building interactive UIs. Moreover, React promotes the use of function components, so there's no syntax overhead of using classes or objects |
| Algorithm for number to word conversion | The time complexity of this algorithm is O(n), where n is the number of digits in the number. The space complexity of the algorithm is O(1), since it uses a fixed amount of extra space regardless of the input size and this algorithm serves its purpose. |
| Using JavaScript instead of TypeScript in this case | JavaScript speeds up the development process. For this project, where the focus is on quick development and deployment of a relatively simple frontend, JavaScript provides sufficient functionality. However, TypeScript could be considered in the future for added type safety and maintainability as the project grows. |