I used a generic linked binary tree to create a Morse Code Converter Utility for my most recent project. Developing a system that could translate Morse code into English text was the primary goal. I did this by using a binary tree structure, in which each node stood for a portion of a Morse code letter or symbol. By going through the tree and assigning the appropriate English letter to each Morse code pattern, I put the conversion logic into practice. I gained a better understanding of the practical uses of binary trees—like decoding Morse code—thanks to this project.

In order to build a versatile tree structure that could effectively store and process the Morse code, I worked with generic classes and TreeNode objects throughout the project. In order for the tree to handle any kind of data, not simply Morse code symbols, I concentrated on making it general. I was able to save the Morse code as a sequence of branching nodes using a linked binary tree, which made it simple to search for the right character. By following the right paths in the tree, I was able to transform a series of dots and dashes into legible English letters.

The ability to use data structures like binary trees in practical situations was the most useful lesson learned from this project. I became more knowledgeable about the applications of trees for search and traversal, two fundamental ideas in computer science. The project also reaffirmed how crucial it is to divide difficult jobs into smaller, more doable phases, including implementing the conversion logic, testing the system, and producing the binary tree. All things considered, this project improved my problem-solving abilities and provided me with practical knowledge with data structures.