The main goal of this thesis was to implement a functional programming (FP) library named Funk that extends C# with support for concepts present in functional programming languages, such as F# and Scala. Funk utilizes many functional programming concepts, including immutability, pattern matching, and various types of monads, together with stronger typing. Introduction of these concepts into C# helps in avoiding many runtime errors and boilerplate code, and it also lets developers write C# code in a declarative rather than in an imperative way, making the day-to-day software development easier and less error-prone. Additionally, the thesis analyzes and compares Funk with existing functional programming libraries such as Language-ext and FuncSharp. Finally, it analyzes the new features of C# 8, which include nullable reference types and pattern matching and compares them with the functionalities of the Funk library.