Experience and Lessons of Developing the Project

Both Elm and Clojure are good options.

On the one hand, Elm has code friendly. It has a nice Haskell-like syntax. Also, Elm moves the Runtime Errors into the Compile Time Errors. Moreover, Elm helps finding bugs more easily and faster in the development cycle. Additionally, Elm is easier to maintain. However, the most challenging part of the project is hard for Elm to install packages and extensions. We find that it is not enough books and resources to learn the installation.

On the other hand, Clojure is much closer to Lisp syntax. It is more mature because Clojure has a lot of libraries, mush more tools and frameworks, and good performance. And there are amounts of books and other resources to learn Clojure. Additionally, Clojure is dynamically typed and compiles with the JVM.

In addition, Clojure is more like a replacement for Java Script, whereas Elm is more like a replacement for HTML, CSS, and Java Script. Both of those are extra requirements for a Node.js.