Documentation

React/TypeScript + Hooks + Firebase project that handles the problem of number splitting into smaller parts without duplicates and zero values.

StyledComponents and SASS were used for styling.
ContextAPI was implemented for state management.
We used React-Hook-Form for validation of the input field.
Formatting configuration defined with ESLint and Prettier.
Working on JEST and React Testing library skills through this project.

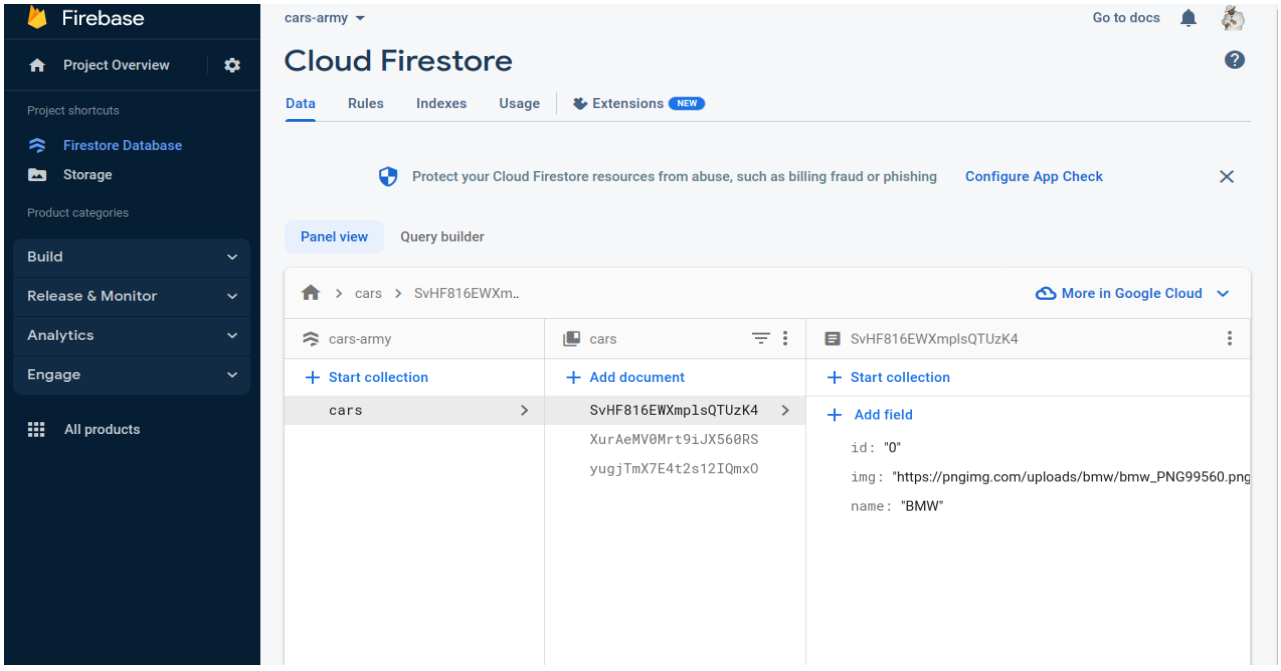
The project is deployed to Netlify, and the demo can be seen here.

Back-end

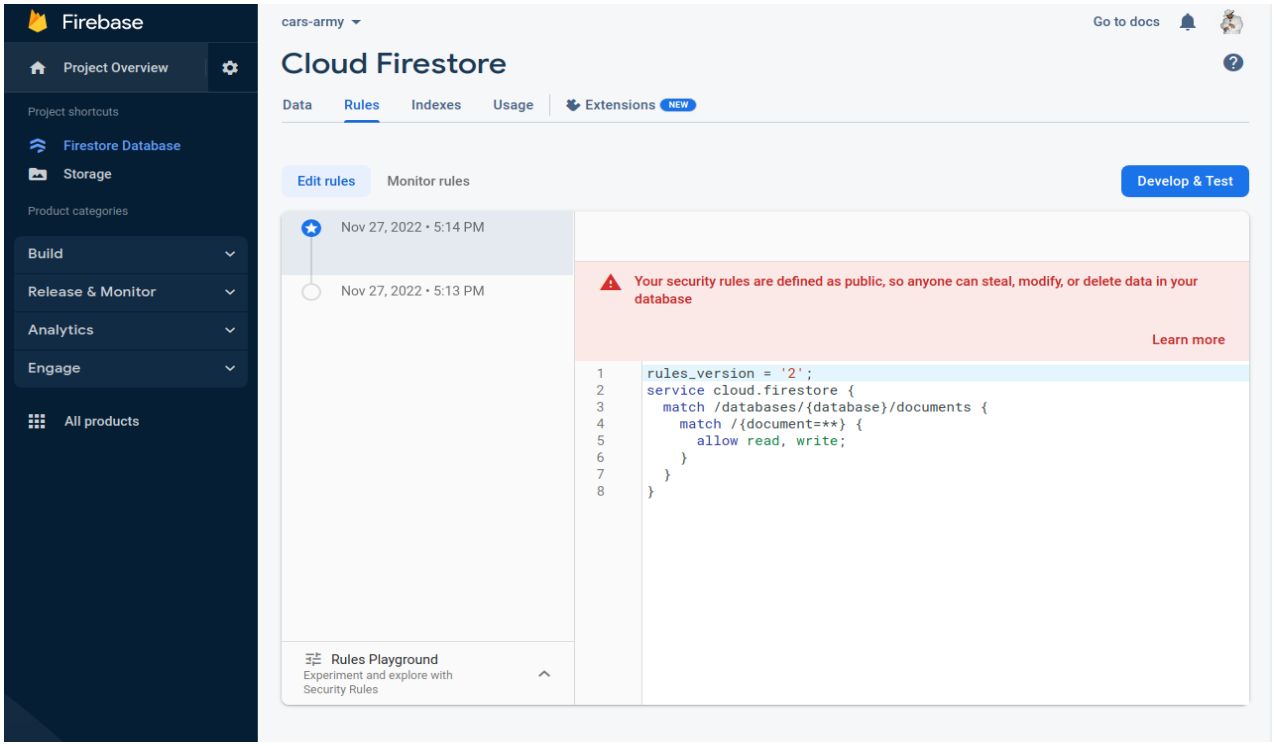
Because this was a relatively simple problem, we used Google Firebase as a back-end service and we used its Cloud Firestore for the database.

- Email: chadbutlerc@gmail.com
- Password: Lilili11!

Database is called cars-army and looks like this:



Currently, it's allowed to access and modify data in there by anyone because rules are set for testing purposes only!



How it works

The logic behind the application and manipulation of the input number is done through *'GlobalContext'* with the use of the data from the '*FirebaseContext'* by calling the util function *'calculateParts*' that is creating random numbers with the help of the '*randomNum'* function and checking with an array of previous results to return non-duplicate/non-zero results.

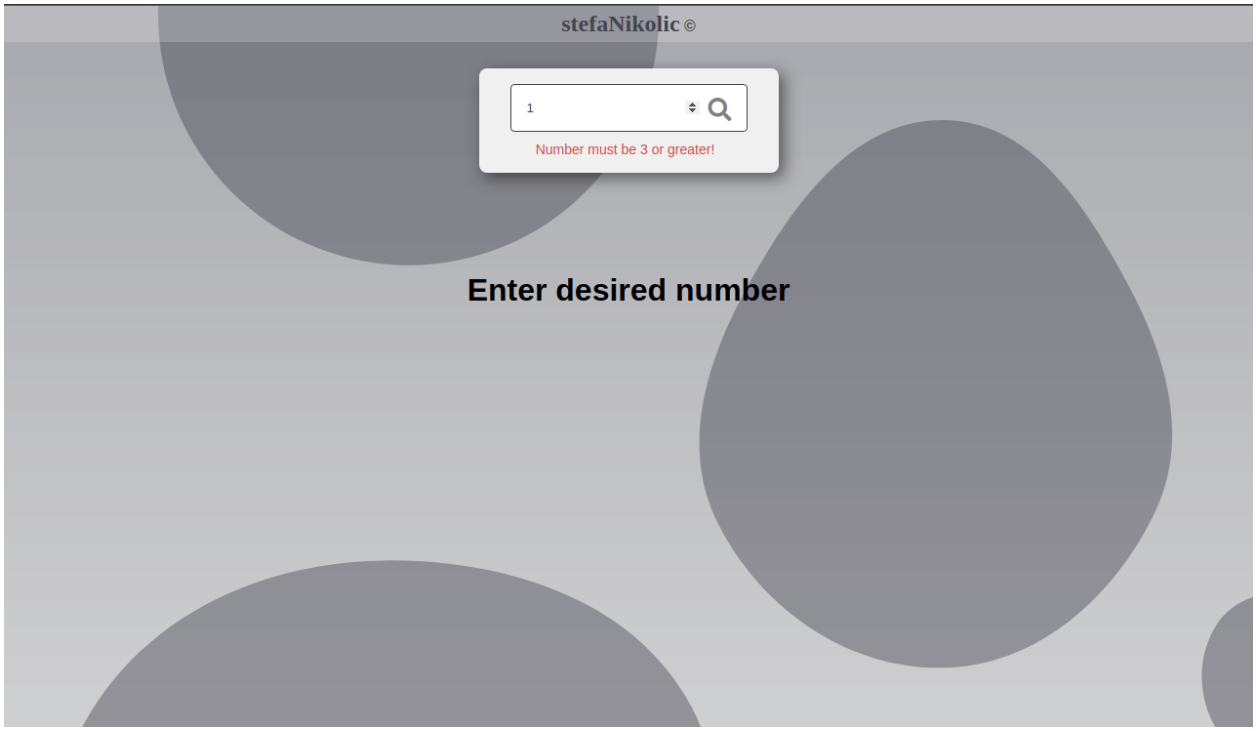
Idea and Usage

Idea is that when the user enters desired number(must be higher than the number of Cars we have in the DB) application splits it into different smaller parts based on the number of cars that we have.

* Images are from Google and used only as an example!

There is error handling for the following cases:

- If the user enters an invalid number(smaller than the number of cars), the application will show an error message



- If the user enters an empty string

Available Scripts

Versions of used packages:

- A version of Node: 16.13.1
- A version of NPM: 8.5.5

In the project directory, you first need to run the: **npm install**
And when packages get installed, you can run the: **npm start**

Runs the app in development mode.

Open http://localhost:3000 to view it in the browser.
The page will reload if you make edits.