Fault In Our Stars | Bug Submission

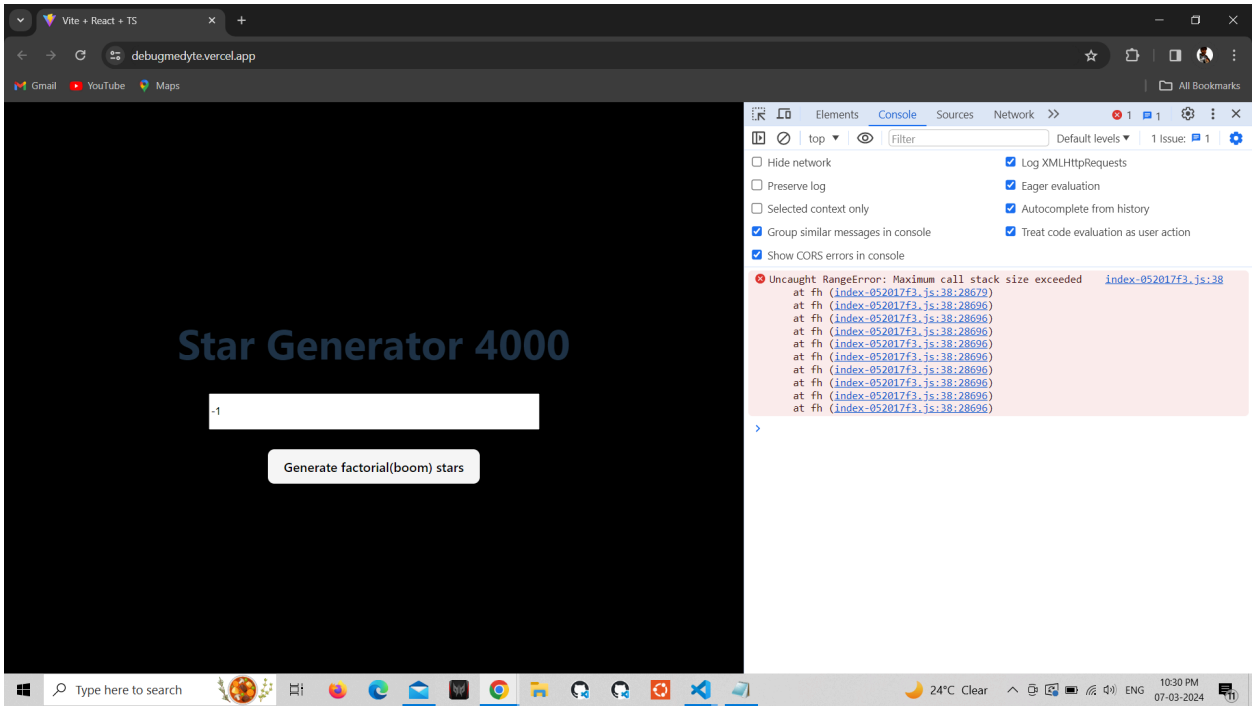
SL. N0 - 1

Title/Description:

Negative Number Input Error

Bug Discovery:

It was observed that when entering a negative number into an input field, an error occurred, leading to the webpage freezing. Further investigation revealed that this error manifested as a "Maximum call stack size exceeded" error in the JavaScript console. This error pointed to a specific function (`fh`) in the `index-052017f3.js` file at line 38.



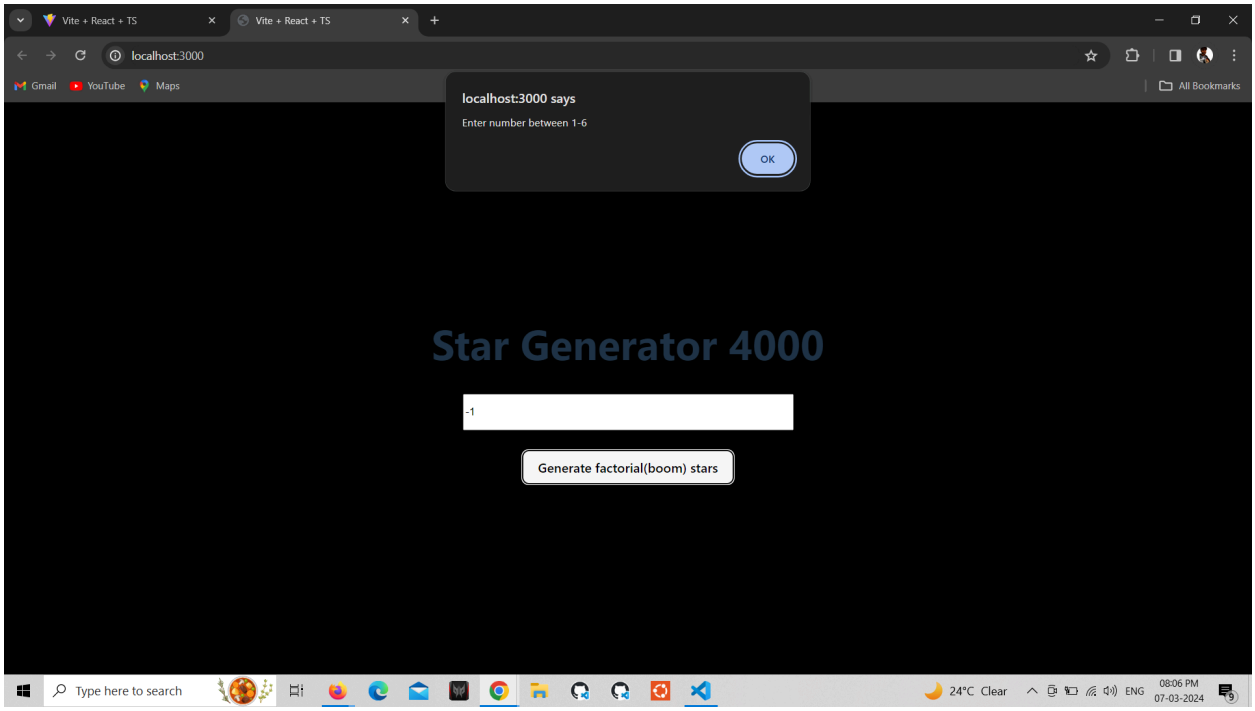
Bug Fixed :

The input `e` is checked to see if it's less than 0 (i.e., negative).

If `e` is negative, an alert is displayed to notify the user to enter a number between 1 and 6.

The function then returns 0 to prevent further processing if the input is negative.

```
function fh(e) {

//Handling exception if value is negative

if (e < 0) {

alert("Enter a number between 1-6")

return 0;

// further code

}}
```



SL. N0 - 2

Title/Description:

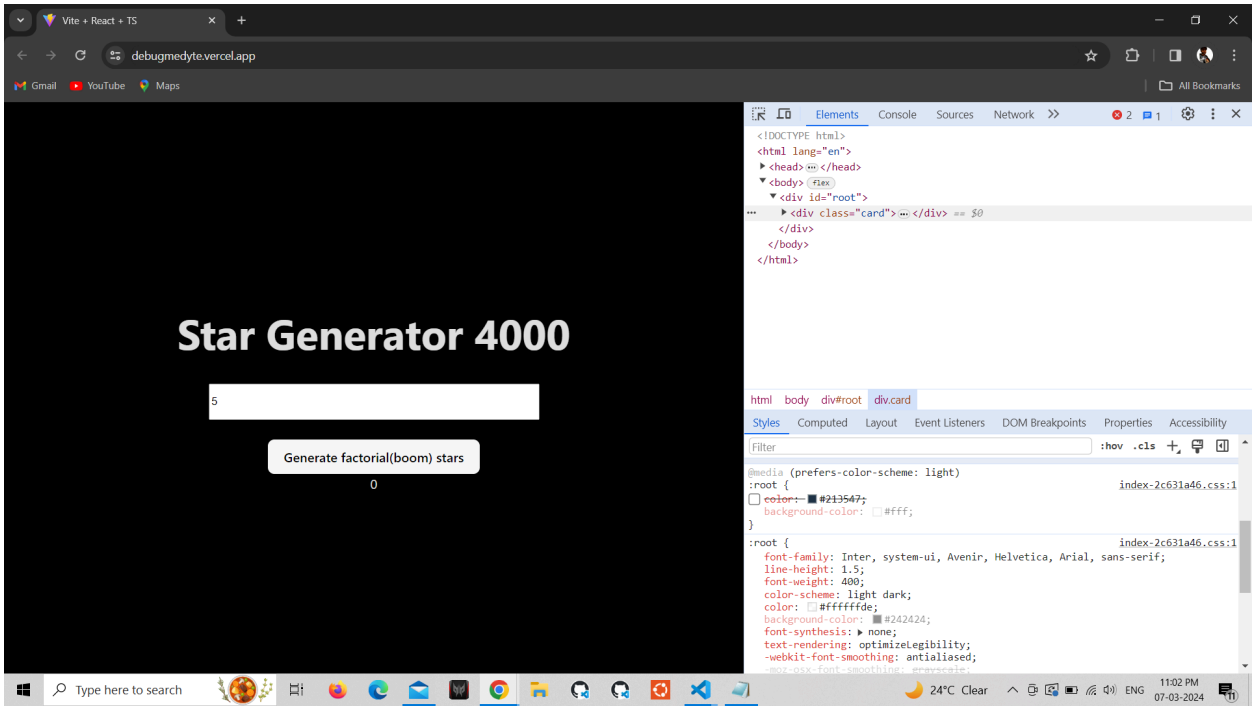
Incorrect Factorial implementation code

Bug Discovery:

```
function fh(e) {
   return 0 === e ? 0 : e * fh(e - 1)
}
```

The code checks if `e` equals 0 and returns 0, implying that the factorial of 0 is 0. However, this contradicts the definition of factorial, where the factorial of 0 is actually 1.

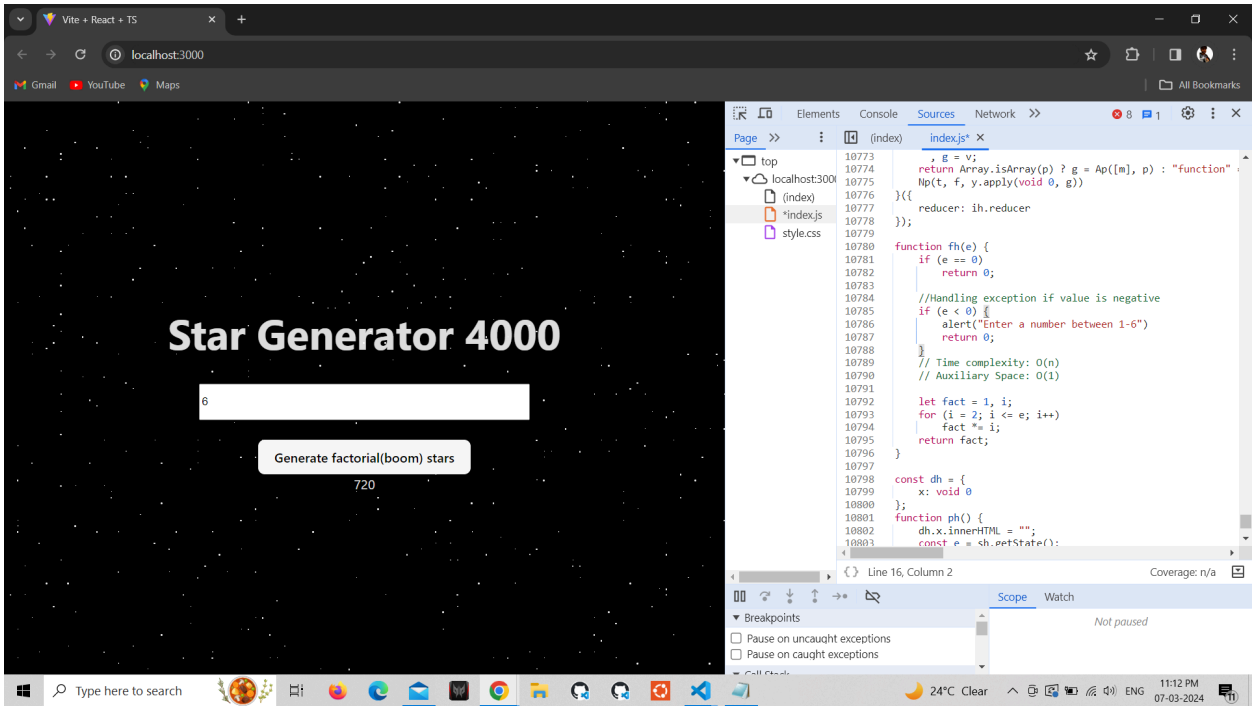
Additionally, the recursive part `e * fh(e - 1)` is used to compute the factorial of `e`. While this recursive approach is conceptually correct, it will produce 0 for all input ranges.


Vite + React + TS
debugmedyte.vercel.app
Gmail
YouTube
Maps
All Bookmarks
Star Generator 4000
5
Generate factorial(boom) stars
0
Elements
Console
Sources
Network
<!DOCTYPE html>
<html lang="en">
<head> ⋯ </head>
<body> flex

 ⋯ == $0

</body>
</html>
html
body
div#root
div.card
Styles
Computed
Layout
Event Listeners
DOM Breakpoints
Properties
Accessibility
Filter
:hov .cls
@media (prefers-color-scheme: light)
:root {
color: #213547;
background-color: #fff;
}
:root {
font-family: Inter, system-ui, Avenir, Helvetica, Arial, sans-serif;
line-height: 1.5;
font-weight: 400;
color-scheme: light dark;
color: #ffffffde;
background-color: #242424;
font-synthesis: none;
text-rendering: optimizeLegibility;
-webkit-font-smoothing: antialiased;
Type here to search
24°C Clear
ENG
11:02 PM
07-03-2024

Bug Fixed:


Vite + React + TS
localhost:3000
Gmail
YouTube
Maps
All Bookmarks
Star Generator 4000
6
Generate factorial(boom) stars
720
Elements
Console
Sources
Network
Page
(index)
index.js*
top
localhost:300
(index)
*index.js
style.css
, g = v;
return Array.isArray(p) ? g = Ap([m], p) : "function"
Np(t, f, y.apply(void 0, g))
}({
reducer: ih.reducer
});
function fh(e) {
if (e == 0)
return 0;
//Handling exception if value is negative
if (e < 0) {
alert("Enter a number between 1-6")
return 0;
}
// Time complexity: O(n)
// Auxiliary Space: O(1)
let fact = 1, i;
for (i = 2; i <= e; i++)
fact *= i;
return fact;
}
const dh = {
x: void 0
};
function ph() {
dh.x.innerHTML = "";
Line 16, Column 2
Coverage: n/a
Scope
Watch
Not paused
Breakpoints
Pause on uncaught exceptions
Pause on caught exceptions
Type here to search
24°C Clear
ENG
11:12 PM
07-03-2024

```
function fh(e) {     if (e == 0)  return 0;

   //Handling exception if value is negative

   if (e < 0) {   alert("Enter a number between 1-6");

    return 0;   }

   // Time complexity: O(n)

   // Auxiliary Space: O(1)

let fact = 1, i;

 for (i = 2; i <= e; i++)  fact *= i;

   return fact; }
```

Description:

- The function first checks if the input `e` is equal to 0. If so, it returns 0. This condition appears incorrect because the factorial of 0 is actually 1, not 0. However, for the purposes of this explanation, let's assume this behavior is intentional.
- Next, the function checks if the input `e` is negative. If it is, an alert is shown to the user, prompting them to enter a number between 1 and 6, and the function returns 0.
- After error checking, the function initializes the `fact` variable to 1.
- It then iterates from 2 to `e` using a `for` loop, calculating the factorial incrementally.
- Within each iteration, the `fact` variable is multiplied by the current value of `i`.
- Finally, the calculated factorial value is returned.

Time Complexity: The time complexity of this function is O(n), where n is the value of `e`, because it iterates from 2 to `e`.

Space Complexity: The space complexity is O(1) because the function uses only a constant amount of space regardless of the input size.

Submitted By

Aashish Kumar Singh

ashish.kumar.singh.jee@gmail.com