# Exercise 2: Solutions + Shoutouts | Complete React Course in Hindi #20

In the last tutorial, we have created our amazing TextUtils application and have successfully deployed it. In this tutorial, we will be solving the exercise which was assigned to you in the 14th tutorial of this React course.

**Note:** We are just doing the following changes to our application just to learn something new and hence would not be deploying it in our mytextutils.com application.
Let's remove the Enable Dark Mode button instead of that we would be creating a color palette, which would change the background color of our application.

# Creating A Color Palette for TextUtils

Follow the below steps to create the color palette for the TextUtils application:

## 1. Editing Toggle Mode Function

If you remember toggle mode was the function, which was responsible for switching between dark and light mode and we have assigned it to our 'Enable Dark Mode' Switch. We need to make some changes to that function so that while clicking on the desired palette it changes the background of the application to that specific color.

## Toggle Mode Function:

```
const toggleMode = (cls) => {
    console.log(cls)
    document.body.classList.add('bg-' + cls)
    if (mode === 'light') {
        setMode('dark');
        document.body.style.backgroundColor = '#042743';
        showAlert("Dark mode has been enabled", "success");
    }
    else {
        setMode('light');
        document.body.style.backgroundColor = 'white';
        showAlert("Light mode has been enabled", "success");
    }
}
```

**Explanation:** We have assigned a class(‘cls’) parameter in the toggle mode function. ‘cls’ will be the value that we want to set our mode as. Suppose, you click on the primary button having this function, then this function will be invoked with the ‘cls’ value as primary. This means that the background color will be ‘blue’ and the ‘primary’ keyword will be printed in the console, as we have used ‘cls’ with them.

# Creating a Custom palette

We would be creating the custom palettes in the Navbar so let's open Navbar.js and add some buttons. We are adding blue, Yellow, Red, Green buttons as:

**Blue palette:**

```
<div className="bg-primary rounded mx-2" onClick={props()=>{props.toggleMode('primary')}} sty
```

**Green palette:**

```
<div className="bg-success rounded mx-2" onClick={props()=>{props.toggleMode('success')}} sty
```

**Red palette:**

```
<div className="bg-danger rounded mx-2" onClick={props()=>{props.toggleMode('danger')}} style
```

**Yellow palette:**

```
<div className="bg-warning rounded mx-2" onClick={props()=>{props.toggleMode('warning')}} sty
```

Similarly, You can create a palette of your desired Colors.

**Explanation:** In the above code, We have assigned the ToggleMode function to the Onclick properties of the buttons.

**For example**, On clicking the primary button the Toggle Mode function will be invoked with the 'cls' value as primary. Remember we have used 'cls' in console.log() and also with background color, so the background color of the application will turn 'Blue', and the 'primary' keyword will be printed in the console.
In a similar fashion, the toggle mode function will be invoked for other keywords, such as warning, danger, success, etc, as well.

**Result:** Now, On clicking the button having a specific color changes the background of our application to that color.

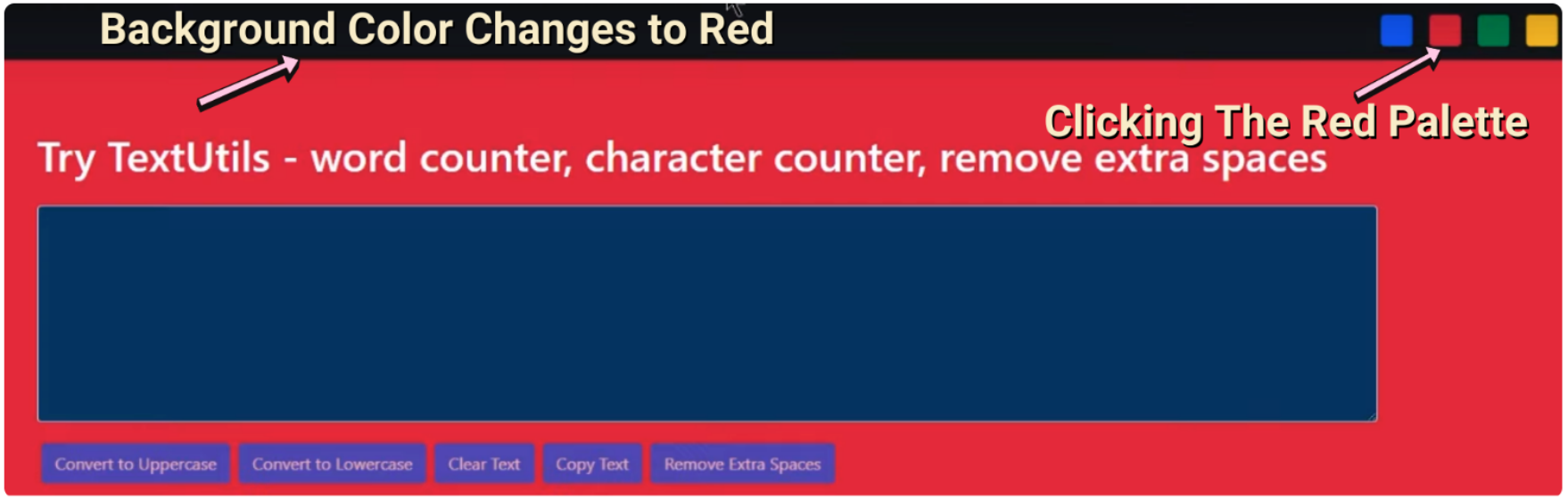


**Figure 1.1: Result**

# Remove Body Classes

**Bug:** You might have noticed that you are unable to use the complete palette without reloading the page. This issue occurs as once we have entered our class('cls') to the function, that is we have clicked on any palette, then a new background class('cls') will be ignored until and unless the old class is removed.

**Solution:** To solve this issue we would create a function removeBodyClasses. This function will remove all the classes from the document.body. The function is as follows:

```
const removeBodyClasses = ()=>{
    document.body.classList.remove('bg-light')
    document.body.classList.remove('bg-dark')
    document.body.classList.remove('bg-warning')
    document.body.classList.remove('bg-danger')
    document.body.classList.remove('bg-success')
}
```

Now, we would Run this function while toggling mode.

```
const removeBodyClasses(){
  document.body.classList.remove('bg-light')
  document.body.classList.remove('bg-dark')
  document.body.classList.remove('bg-warning')
  document.body.classList.remove('bg-danger')
  document.body.classList.remove('bg-success')
}

const toggleMode = (cls)=>{
  removeBodyClasses()
  console.log(cls)
  document.body.classList.add('bg-'+cls)
  if(mode === 'light'){
    setMode('dark');
    document.body.style.backgroundColor = '#042743';
    showAlert("Dark mode has been enabled", "success");
  }
  else{
    setMode('light');
    document.body.style.backgroundColor = 'white';
    showAlert("Light mode has been enabled", "success");
    // document.title = 'TextUtils - Light Mode';
  }
}
```

Remove Body Classes function

Running the above function while Toggling Modes

# Our Application

Here is the demo of our Color Palette. On clicking the Red palette, the background of our application changes to Red.

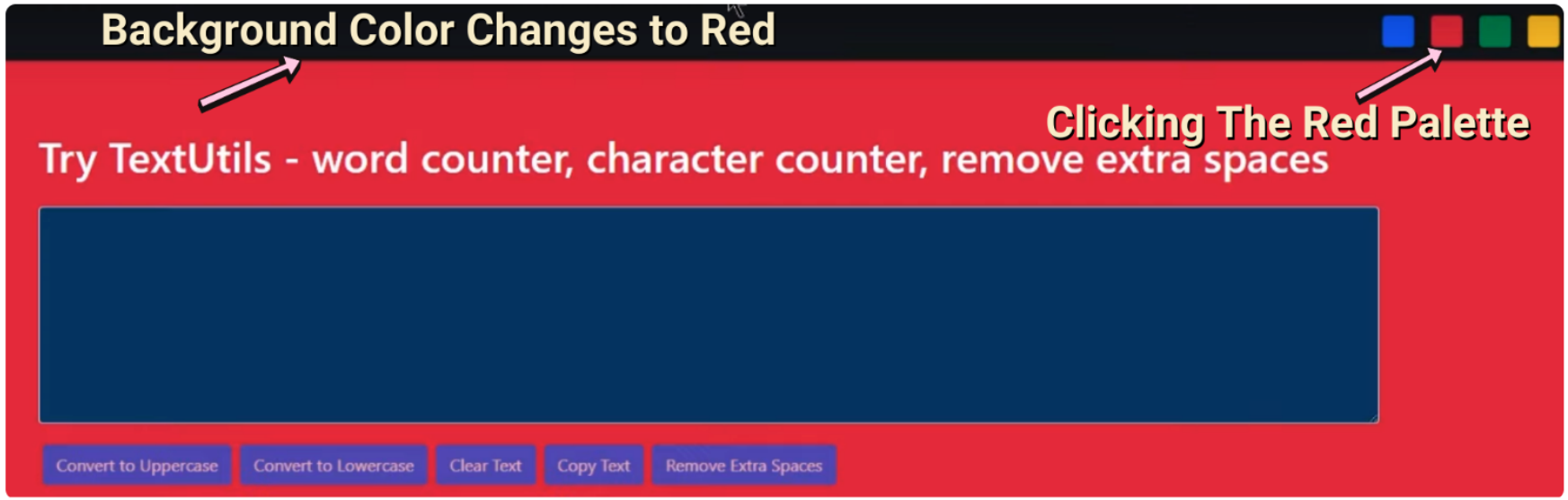


Figure 1.3: Our Color Palette

**Hence, we have successfully created our color palette.**

So here we complete the 20th video of this React series. In the upcoming videos, we will be covering a lot of new concepts of React with some new and interesting projects that would allow you to understand React completely.

Do remember to Access the playlist of the Free React Course in Hindi by clicking here.

Check out our Github to access the source code at this point in the course, Click Here

**Thank you for being with me throughout the tutorial. In the next tutorial, we will be fixing a few more TextUtils issues. Till then, Keep learning and Keep Coding.**