**Digital Nurture 7.0 Deep Skilling**

**Week 7**

**Hands on 1**

Create a React Application named “cricket app” with the following components:

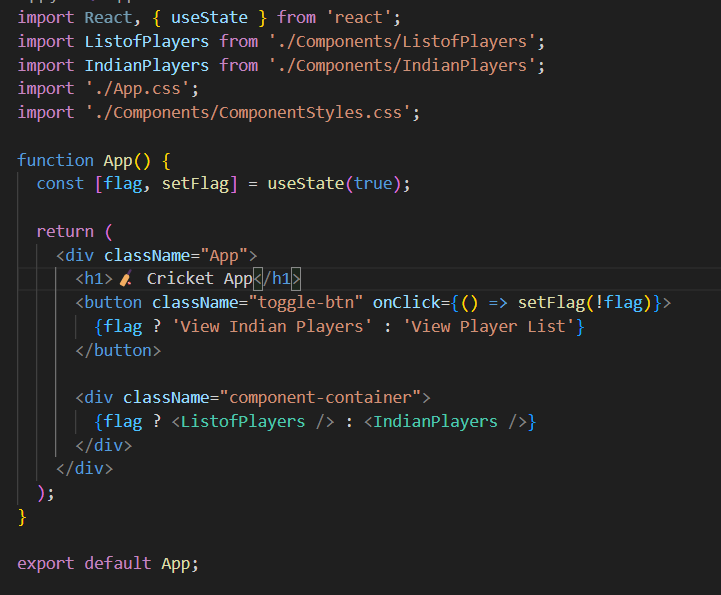
1. List of Players

* Declare an array with 11 players and store details of their names and scores using the map feature of ES6
* Filter the players with scores below 70 using arrow functions of ES6.

2. Indian Players

1. Display the Odd Team Player and Even Team players using the Destructuring features of ES6
2. Declare two arrays T20players and Ranji Trophy players and merge the two arrays and display them using the Merge feature of ES6

Display these two components in the same home page using a simple if else in the flag variable.

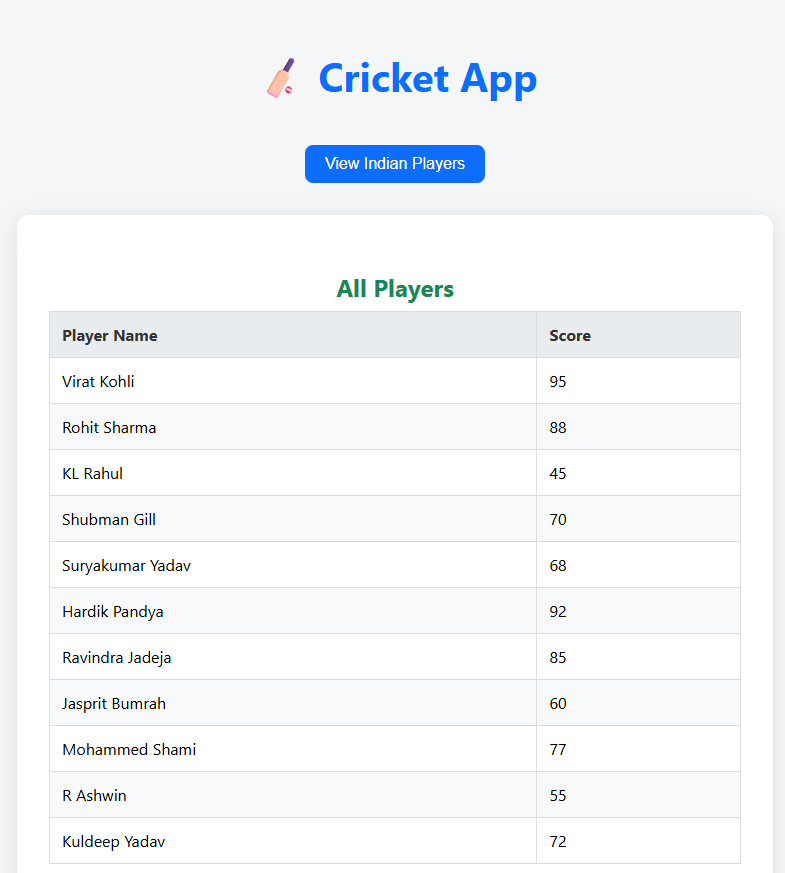
. 

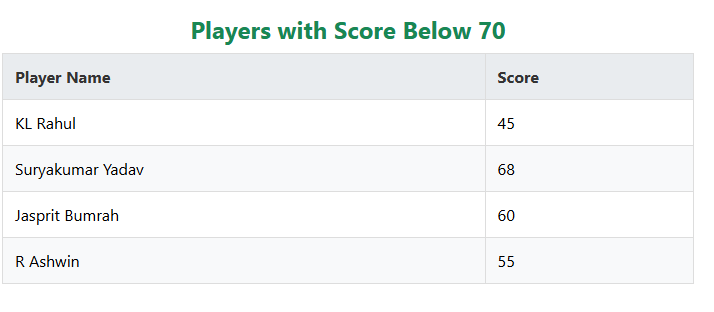




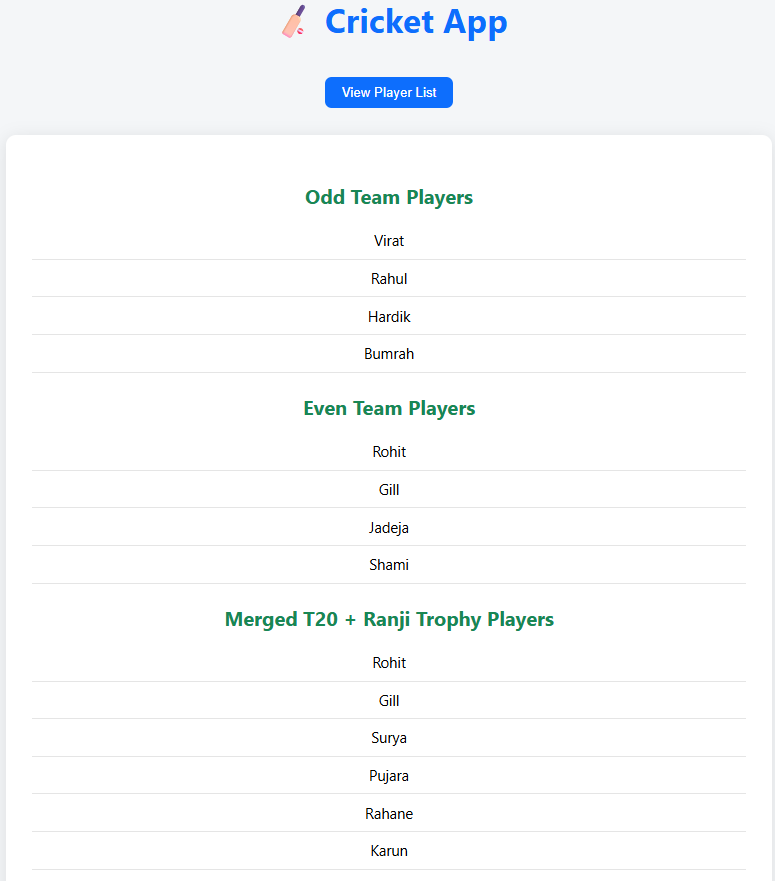
**Output :**

When flag = true

****

****

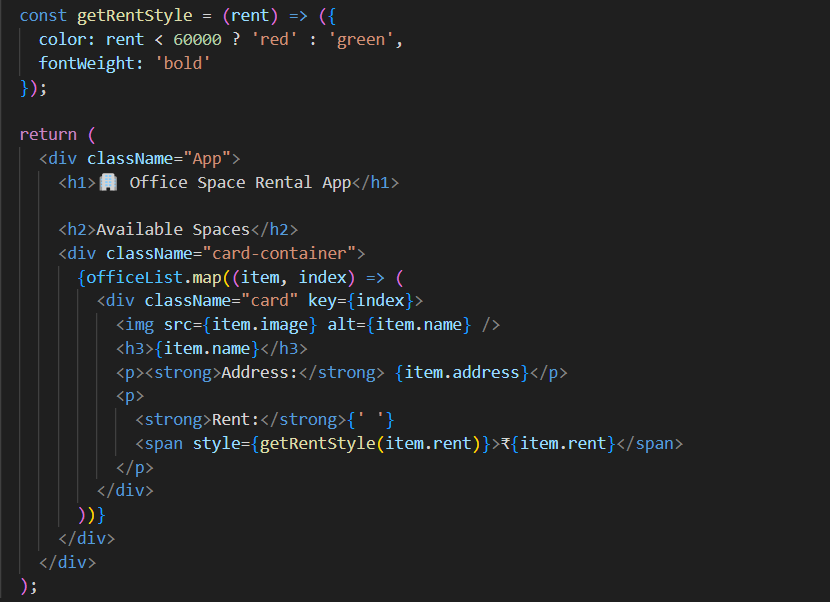
When flag=false

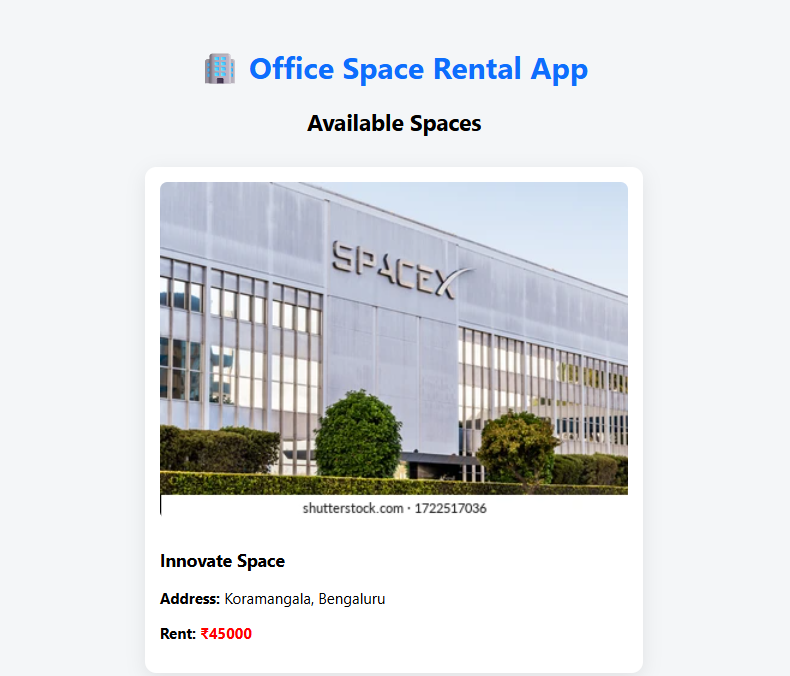
****

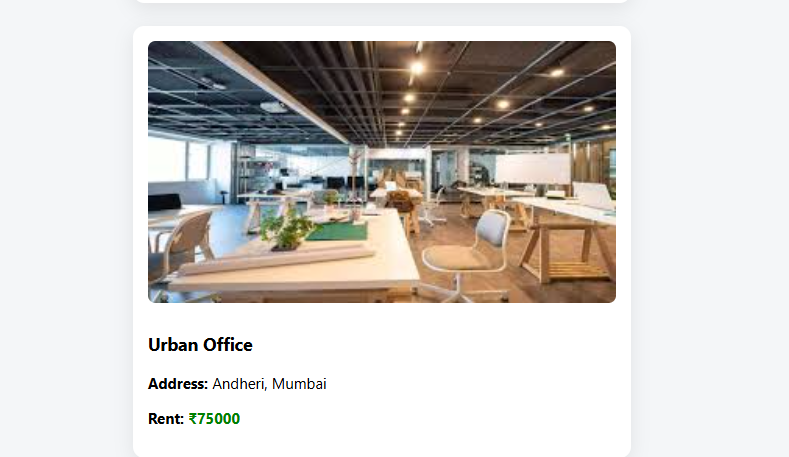
**Hands on 2**

Create a React Application named “office space rental app” which uses React JSX to create elements, attributes and renders DOM to display the page.

1. Create an element to display the heading of the page.
2. Attribute to display the image of the office space
3. Create an object of office to display the details like Name, Rent and Address.
4. Create a list of Object and loop through the office space item to display more data.
5. To apply css, Display the color of the Rent in Red if it’s below 60000 and in Green if it’s above 60000.

****

**Output :**

****

Hands on 3

Create a React application named **"event examples app"** to handle various form events.

 Include an **"Increment"** button to increase a counter and a **"Decrement"** button to decrease it.

 The "Increment" button should invoke multiple methods:

* One method to increment the counter value.
* Another method to display a static message like **"Hello Member1"**.

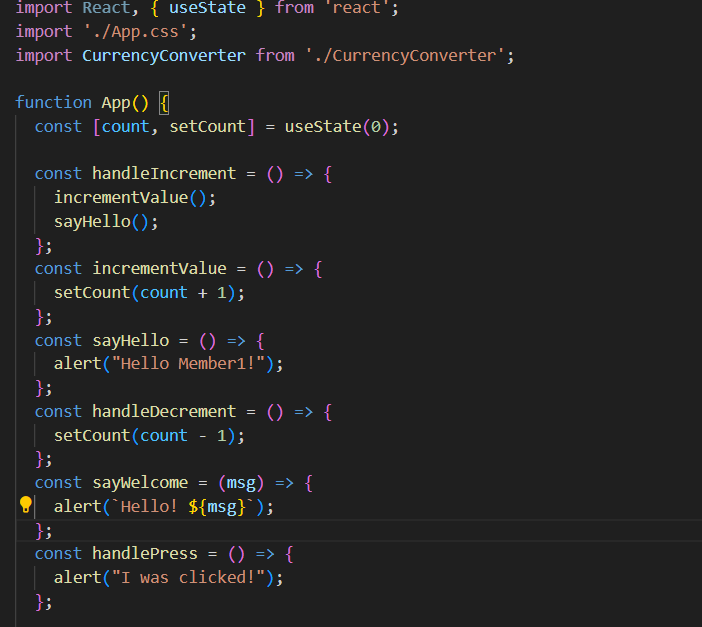
 Create a **"Say Welcome"** button that invokes a function taking **"Welcome"** as an argument and displays it.

 Add a **"Click on me"** button that demonstrates a **synthetic event**, showing the message **"I was clicked!"** when triggered.

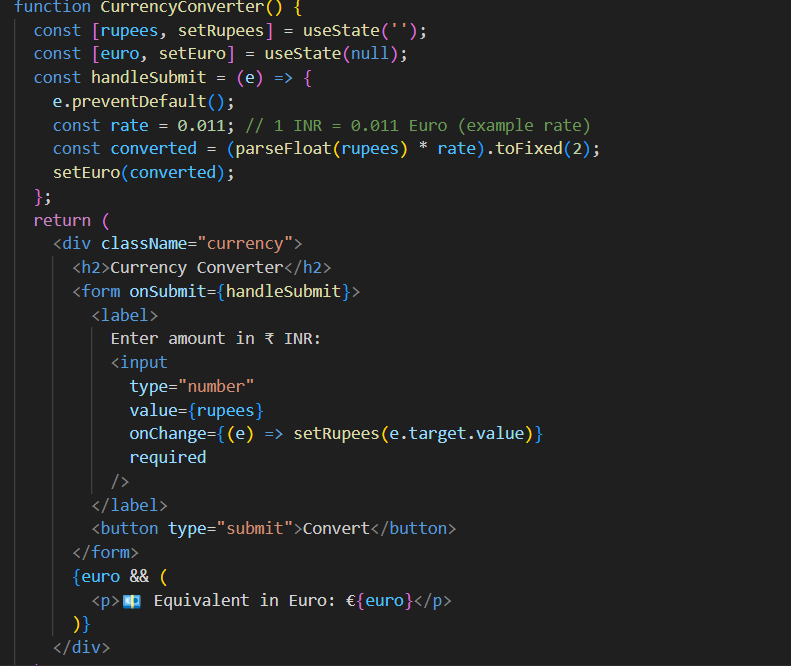
 Create a separate component named **"Currency Convertor"**.

* This component should include two input fields: one for **amount in Indian Rupees** and another for the **target currency** (e.g., Euro).
* Include a **"Submit"** button that, when clicked, triggers the **handle Submit** event.
* The event handler should simulate or alert a conversion of the given amount from rupees to the specified currency.

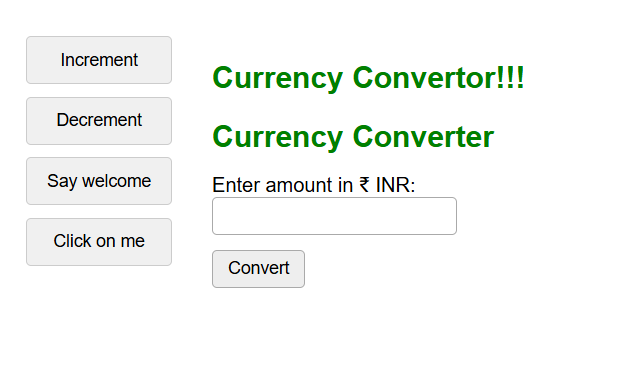
Code :

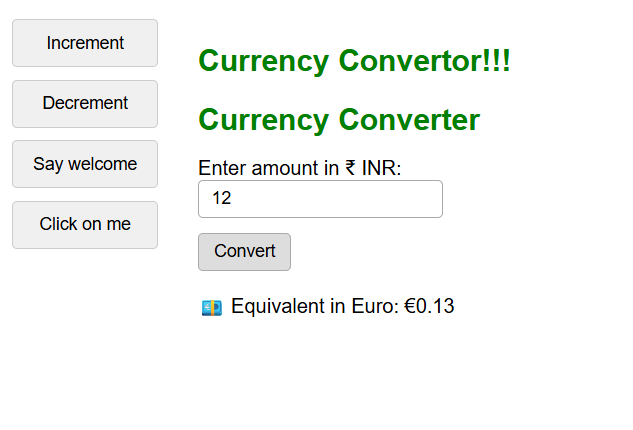


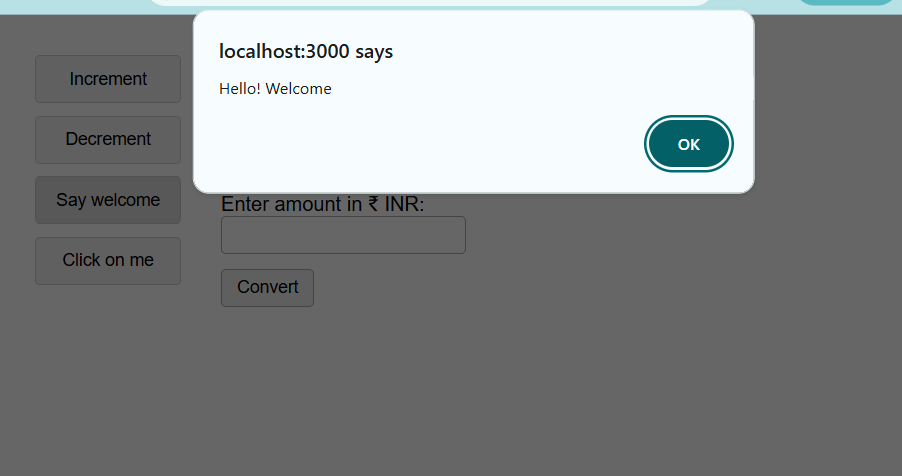


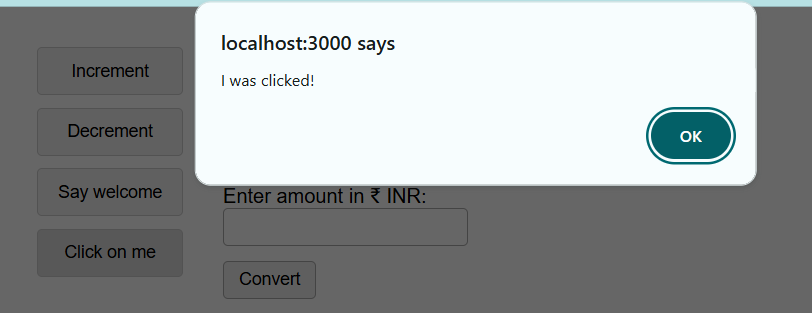
****

**Output :**

****

****

****

****

Hands on 4

Create a React Application named “ticket booking app” where the guest user can browse the page where the flight details are displayed whereas the logged in user only can book tickets.

The Login and Logout buttons should accordingly display different pages. Once the user is logged in the User page should be displayed. When the user clicks on Logout, the Guest page should be displayed.

**Code:**

**A screen shot of a computer program

Description automatically generated**

**A screen shot of a computer screen

Description automatically generated**

**A screen shot of a computer program

Description automatically generated**

**Output :**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Hands on 5**

Estimated time to complete this lab: **60 minutes.**

Create a React App named “bloggerapp” in with 3 components.

1. Book Details
2. Blog Details
3. Course Details

Implement this with as many ways possible of Conditional Rendering.

**Code :**

**A screen shot of a computer program

Description automatically generated**

**A screen shot of a computer program

Description automatically generated**

**A screen shot of a computer program

Description automatically generated**

**A screen shot of a computer program

Description automatically generated**

**A screen shot of a computer program

Description automatically generated**

**Output :**

**A black screen with white text

Description automatically generated**