React Handson
Building Components
project name: react-basics

In this project react-basics we will start from the Create React APP (CRA) – rename it to react-classes, build several more routes in the views folder and use them in the rootCMP with router V5 + Switch.

Challenge #1 - animal-list component

This component receive a prop: animalInfos and render a list of animals and their count:

Here is an array:

```
animalInfos = [
        {type: 'Malayan Tiger', count: 787},
        {type: 'Mountain Gorilla', count: 212},
        {type: 'Fin Whale', count: 28},
    ]
```

The search link opens a new tab with a search for such animal in google
(e.g.: `https://www.google.com/search?q=Malayan Tiger`)

Challenge #2 –season-clock component

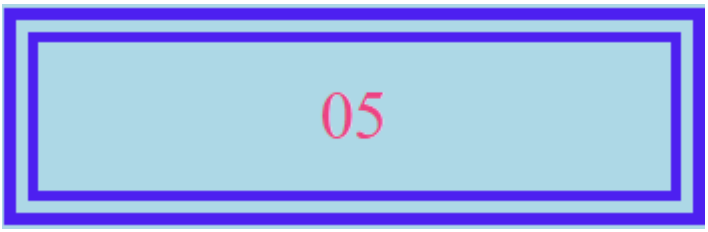
This component shows the current season, month name and day name (use the provided functions in utilService), when clicking the component, its background should toggle between dark and light colors

Component initial state:

```
state = {
   clock: new Date(),
   isDark: true
}
```

- Show the current season 
- Add a running clock - use an interval to update the state every second, remember to clear the interval when component is unmounting

Challenge #3 – count-down component



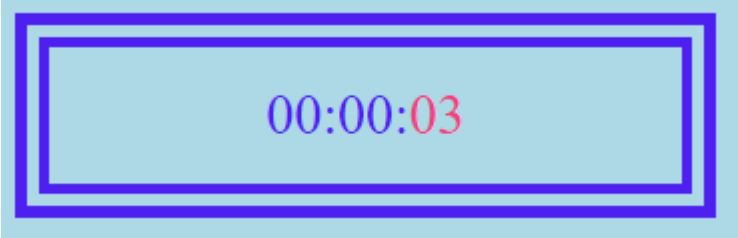
This component has 2 props: *startFrom* and *onDone*.
The component shows a counter going down from startFrom to 0
This is how you will use that component for counting 10 seconds:

```
<CountDown startFrom={100} onDone={()=>{
    console.log('Done!');
}} />
```

- During the last 6 seconds, the seconds should make in Red
- When time ends call the onDone function

Bonus

if the component receives a toTime property (timestamp), it ignores the startFrom prop and renders a timer going down to the *toTime* prop

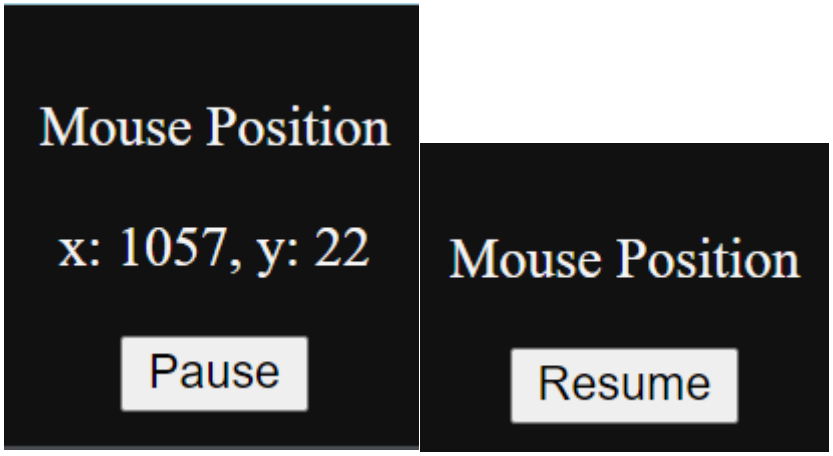


This is how you will use that component for counting 10 seconds:

```
<CountDown toTime={Date.now() + 1000*10} onDue={()=>{
    console.log('Its Time!');
}} />
```

Extra Bonus: in the onDone function play an audio file (mp3)

Challenge #4 – mouse-monitor component



This component renders a section at the bottom-right of the page constantly showing the current position of the mouse.

When Pause is clicked, the section no longer monitors the mouse (you will need to remove the event listener)

The initial state of the component:

```
    isOn : true,
    pos: {
        x: 0,
        y: 0
    }
```

TIP

```
addMouseListener(){
   document.addEventListener('mousemove', this.updatePos)
}
```

Challenge #5 - watcher-app

Build a full CRUDL app using classes (where needed) + Router V5 + Redux + **THUNK**
Use the async-local-storage we already know to make your app asynchronous – with async await etc..

Watcher data model:

```
{
    id: 'w101',
    fullName : 'Puki Ba',
    movies: ['Rambo', 'Rocky']
}
```

Store watchers module initial state:

```
{
   watchers: [],
   selectedWatcher: null
}
```

<WatcherIndex />

This page renders a watchers array (retrieved asynchronously from a service) and allows adding, removing and selecting a watcher.
For adding a user, start simple and use the prompt() function
(Bonus: improve to a watcher edit page **<WatcherEdit />**)
Remember to split into CMPS (WatcherList, Preview etc..)

<WatcherDetails />

When user is selected – render a page showing the user name and his movies list, with a go back button

(Route - /watcher/:watcherId)