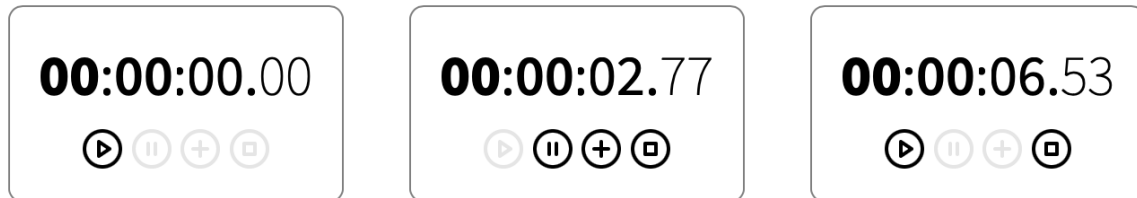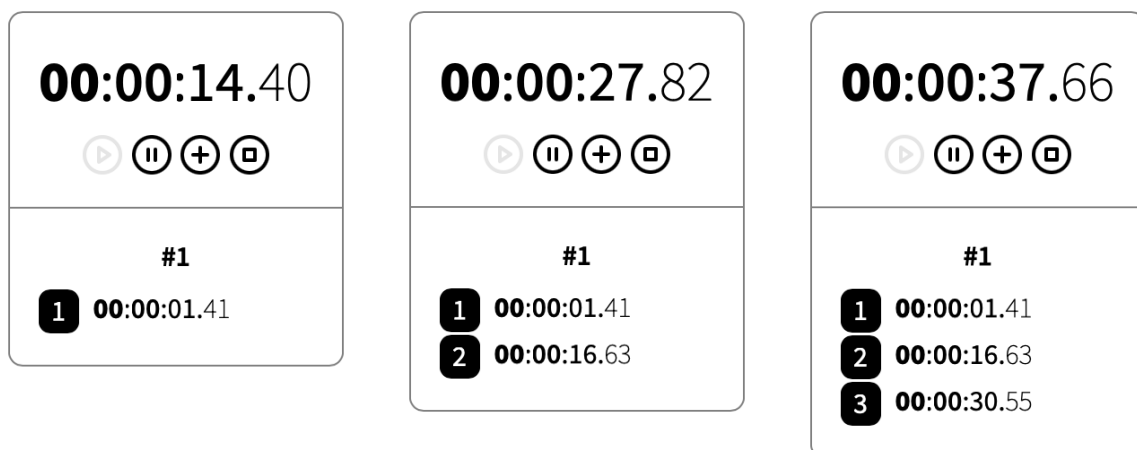# CMPS 350 · Web Development Fundamentals

Tutorial 06 · Lap Timer

Develop a lap timer that can track multiple runs and laps using React and Next.

1. Create a `Timer` component that is composed of a `Panel` component and a `Runs` component. The `Panel` component comprises a `Display` and a `Controls` component. The `Runs` component is composed of multiple `Run` components, tracking multiple laps.

2. The `Display` component renders the elapsed time. It receives a single prop, `time`, that contains the time elapsed in milliseconds, and renders the hours, minutes, seconds, and centiseconds.

3. The `Controls` component renders buttons to control these actions: Start/Resume, Pause, Lap, and Stop/Reset. Each of these actions will have an associated callback to handle its `click` event.

4. The `Run` component renders the laps that have been tracked for a certain run using a `Laps` component. Each `Laps` component is comprised of multiple `Lap` components.

5. Create the required state variables using the `useState` hook. How many are needed?

6. Create a reference, using the `useRef` hook, to be able to clear a set interval. When should the reference hook be used instead of the state hook?

7. ~~Defer the value of the displayed time using the `useDeferredValue` hook.~~

8. Serialize the `Timer` component and save its state in local storage so it can be reloaded between sessions. How can an effect be used to automate the serialization using the `useEffect` hook?

9. Update your `Timer` component so that multiple timers can be created and serialized.

10. An interactive demo of this tutorial is available at https://cmps350-lap-timer.vercel.app.