● ● ●   🔲 index.html

```html
<div class="wrapper">
  <div class="timer-display">00:00:00</div>
  <div class="container">
    <div class="inputs">
      <input type="number" id="hourInput" placeholder="00" min="0" max="23" />
      <input type="number" id="minuteInput" placeholder="00" min="0" max="59" />
    </div>
    <button id="set">Add Alarm</button>
    <div class="activeAlarms"></div>
  </div>
</div>
```

CSS

```css
* {
  padding: 0;
  margin: 0;
  box-sizing: border-box;
  font-family: "Poppins", sans-serif;
}
body {
  background-color: #4838d6;
}
.hide {
  display: none;
}
.wrapper {
  background-color: #ffffff;
  width: 90%;
  max-width: 31.25em;
  padding: 3.12em 5em;
  position: absolute;
  transform: translateX(-50%);
  left: 50%;
  top: 1em;
  border-radius: 0.5em;
}
.timer-display {
  font-size: 2.18em;
  text-align: center;
  font-family: "Roboto Mono";
}
```

```css
.inputs {
  display: flex;
  align-items: center;
  justify-content: center;
  gap: 1em;
  margin-top: 2em;
}
.inputs input {
  width: 2.8em;
  font-size: 1.3em;
  border: 1px solid #000000;
  border-radius: 0.3em;
  padding: 0.4em 0.2em;
}
#set {
  background-color: #377dff;
  border: none;
  padding: 0.9em 1.8em;
  display: block;
  margin: 1.5em auto 0 auto;
  border-radius: 2em;
  color: #ffffff;
}
.alarm {
  display: grid;
  grid-template-columns: 8fr 2fr 2fr;
  gap: 1em;
  margin-top: 1.5em;
  align-items: center;
  border-bottom: 1px solid #898f9b;
  padding-bottom: 0.6em;
}
```

```css
.alarm input[type="checkbox"] {
  appearance: none;
  height: 2em;
  width: 3.75em;
  background-color: #e2e2ec;
  border-radius: 1.25em;
  position: relative;
  cursor: pointer;
  outline: none;
}
.alarm input[type="checkbox"]:before {
  position: absolute;
  content: "";
  background-color: #757683;
  height: 1.43em;
  width: 1.43em;
  border-radius: 50%;
  top: 0.25em;
  left: 0.25em;
}
.alarm input[type="checkbox"]:checked {
  background-color: #d2e2ff;
}
.alarm input[type="checkbox"]:checked:before {
  background-color: #377dff;
  left: 2em;
}
.deleteButton {
  background-color: transparent;
  font-size: 1.5em;
  color: #377dff;
  border: none;
  cursor: pointer;
}
```

```javascript
//Initial References
let timerRef = document.querySelector(".timer-display");
const hourInput = document.getElementById("hourInput");
const minuteInput = document.getElementById("minuteInput");
const activeAlarms = document.querySelector(".activeAlarms");
const setAlarm = document.getElementById("set");
let alarmsArray = [];
let alarmSound = new Audio("./alarm.mp3");

let initialHour = 0,
  initialMinute = 0,
  alarmIndex = 0;

//Append zeroes for single digit
const appendZero = (value) => (value < 10 ? "0" + value : value);

//Search for value in object
const searchObject = (parameter, value) => {
  let alarmObject,
    objIndex,
    exists = false;
  alarmsArray.forEach((alarm, index) => {
    if (alarm[parameter] == value) {
      exists = true;
      alarmObject = alarm;
      objIndex = index;
      return false;
    }
  });
  return [exists, alarmObject, objIndex];
};

//Display Time
function displayTimer() {
  let date = new Date();
  let [hours, minutes, seconds] = [
    appendZero(date.getHours()),
    appendZero(date.getMinutes()),
    appendZero(date.getSeconds())
  ];
```

```javascript
  //Display time
  timerRef.innerHTML = `${hours}:${minutes}:${seconds}`;

  //Alarm
  alarmsArray.forEach((alarm, index) => {
    if (alarm.isActive) {
      if (`${alarm.alarmHour}:${alarm.alarmMinute}` === `${hours}:${minutes}`) {
        alarmSound.play();
        alarmSound.loop = true;
      }
    }
  });
}


const inputCheck = (inputValue) => {
  inputValue = parseInt(inputValue);
  if (inputValue < 10) {
    inputValue = appendZero(inputValue);
  }
  return inputValue;
};


hourInput.addEventListener("input", () => {
  hourInput.value = inputCheck(hourInput.value);
});


minuteInput.addEventListener("input", () => {
  minuteInput.value = inputCheck(minuteInput.value);
});

//Create alarm div

const createAlarm = (alarmObj) => {
  //Keys from object
  const { id, alarmHour, alarmMinute } = alarmObj;
  //Alarm div
  let alarmDiv = document.createElement("div");
  alarmDiv.classList.add("alarm");
  alarmDiv.setAttribute("data-id", id);
  alarmDiv.innerHTML = `<span>${alarmHour}: ${alarmMinute}</span>`;
```

```javascript
    let checkbox = document.createElement("input");
    checkbox.setAttribute("type", "checkbox");
    checkbox.addEventListener("click", (e) => {
      if (e.target.checked) {
        startAlarm(e);
      } else {
        stopAlarm(e);
      }
    });
    alarmDiv.appendChild(checkbox);
    //Delete button
    let deleteButton = document.createElement("button");
    deleteButton.innerHTML = `<i class="fa-solid fa-trash-can"></i>`;
    deleteButton.classList.add("deleteButton");
    deleteButton.addEventListener("click", (e) => deleteAlarm(e));
    alarmDiv.appendChild(deleteButton);
    activeAlarms.appendChild(alarmDiv);
};

//Set Alarm
setAlarm.addEventListener("click", () => {
  alarmIndex += 1;

  //alarmObject
  let alarmObj = {};
  alarmObj.id = `${alarmIndex}_${hourInput.value}_${minuteInput.value}`;
  alarmObj.alarmHour = hourInput.value;
  alarmObj.alarmMinute = minuteInput.value;
  alarmObj.isActive = false;
  console.log(alarmObj);
  alarmsArray.push(alarmObj);
  createAlarm(alarmObj);
  hourInput.value = appendZero(initialHour);
  minuteInput.value = appendZero(initialMinute);
});

//Start Alarm
const startAlarm = (e) => {
  let searchId = e.target.parentElement.getAttribute("data-id");
  let [exists, obj, index] = searchObject("id", searchId);
  if (exists) {
    alarmsArray[index].isActive = true;
  }
};
```

```js
//delete alarm
const deleteAlarm = (e) => {
  let searchId = e.target.parentElement.parentElement.getAttribute("data-id");
  let [exists, obj, index] = searchObject("id", searchId);
  if (exists) {
    e.target.parentElement.parentElement.remove();
    alarmsArray.splice(index, 1);
  }
};

window.onload = () => {
  setInterval(displayTimer);
  initialHour = 0;
  initialMinute = 0;
  alarmIndex = 0;
  alarmsArray = [];
  hourInput.value = appendZero(initialHour);
  minuteInput.value = appendZero(initialMinute);
};
```