**TASK 2:-**

**To build a stopwatch web application, you can use HTML, CSS, and JavaScript. HTML is used to structure the elements of the application. By implementing functions for starting, pausing, and resetting the stopwatch, as well as tracking and displaying lap times, users can accurately measure and record time intervals. With these technologies and functionalities, you can create an interactive and user-friendly stopwatch web application.**

**CODE**:-

<!DOCTYPE html>

<html lang=”en”>

<head>

<meta charset=”UTF-8”>

<title>Stopwatch App</title>

<style>

Body {

Font-family: Arial, sans-serif;

Background: #f4f4f4;

Text-align: center;

Padding-top: 100px;

}

.stopwatch {

Background: #fff;

Padding: 30px;

Border-radius: 15px;

Box-shadow: 0 4px 8px rgba(0,0,0,0.2);

Display: inline-block;

}

.time {

Font-size: 48px;

Margin-bottom: 20px;

}

.buttons button {

Margin: 5px;

Padding: 10px 20px;

Font-size: 16px;

Border: none;

Border-radius: 8px;

Cursor: pointer;

Transition: background 0.3s;

}

.buttons button:hover {

Opacity: 0.8;

}

.lap-times {

Margin-top: 20px;

Text-align: left;

}

.lap-times li {

Font-size: 16px;

}

#startBtn { background-color: #28a745; color: white; }

#pauseBtn { background-color: #ffc107; color: white; }

#resetBtn { background-color: #dc3545; color: white; }

#lapBtn { background-color: #007bff; color: white; }

</style>

</head>

<body>

<div class=”stopwatch”>

<div class=”time” id=”display”>00:00:00</div>

<div class=”buttons”>

<button id=”startBtn”>Start</button>

<button id=”pauseBtn”>Pause</button>

<button id=”resetBtn”>Reset</button>

<button id=”lapBtn”>Lap</button>

</div>

<ul class=”lap-times” id=”laps”></ul>

</div>

<script>

Let startTime, elapsedTime = 0, timerInterval;

Function timeToString(time) {

Let diffInHrs = time / 3600000;

Let hh = Math.floor(diffInHrs);

Let diffInMin = (diffInHrs – hh) \* 60;

Let mm = Math.floor(diffInMin);

Let diffInSec = (diffInMin – mm) \* 60;

Let ss = Math.floor(diffInSec);

Let diffInMs = (diffInSec – ss) \* 100;

Let ms = Math.floor(diffInMs);

Let formatted =

`${hh.toString().padStart(2, ‘0’)}:` +

`${mm.toString().padStart(2, ‘0’)}:` +

`${ss.toString().padStart(2, ‘0’)}`;

Return formatted;

}

Function start() {

startTime = Date.now() – elapsedTime;

timerInterval = setInterval(() => {

elapsedTime = Date.now() – startTime;

document.getElementById(“display”).textContent = timeToString(elapsedTime);

}, 100);

}

Function pause() {

clearInterval(timerInterval);

}

Function reset() {

clearInterval(timerInterval);

document.getElementById(“display”).textContent = “00:00:00”;

elapsedTime = 0;

document.getElementById(“laps”).innerHTML = “”;

}

Function lap() {

Const lapTime = timeToString(elapsedTime);

Const lapItem = document.createElement(“li”);

lapItem.textContent = `Lap: ${lapTime}`;

document.getElementById(“laps”).appendChild(lapItem);

}

Document.getElementById(“startBtn”).addEventListener(“click”, start);

Document.getElementById(“pauseBtn”).addEventListener(“click”, pause);

Document.getElementById(“resetBtn”).addEventListener(“click”, reset);

Document.getElementById(“lapBtn”).addEventListener(“click”, lap);

</script>

</body>

</html>

**Output**:-

