Final Project Report — ICT Lab

University Name: FAST NUCES, Karachi
Department: Department of Computer Science
Course: Information and Communication Technology(ICT)
Project Title: Digital-Clock
Submitted By: Devish Kumar (25K-0984) Shaheer Ismail(25K-0676)
Submitted To: Sir Osama Tahir
Semester: Fall 2025
Date: 30-11-2025

Abstract

This project involves developing a web-based Digital Clock and Countdown Timer using HTML, CSS, and JavaScript. The application displays the current time in real-time and provides a user-settable timer. The digital clock updates every second. The countdown timer allows the user to input minutes and seconds, with functions to Start/Pause/Resume and Reset. A notification sound and a subtle screen vibration animation are included to alert the user when the timer hits zero or at every minute mark during the countdown.

Introduction

The Digital Clock and Timer is a simple but functional web application. It serves two main purposes: displaying the current system time and functioning as a versatile countdown timer. The project is implemented using standard front-end web technologies, demonstrating fundamental programming concepts such as time handling, interval functions, DOM manipulation, and user interface design. This type of application is a classic exercise in programming fundamentals, providing practical experience with real-time updates and event handling.

2. Objectives

- To create a functional digital clock that accurately displays the current time (hours, minutes, seconds) and updates every second.
- To implement a countdown timer allowing users to set a time in minutes and seconds.
- To provide Start, Pause/Resume, and Reset functionality for the timer.
- To play a sound notification and display a visual vibration effect when the timer reaches zero.
- To use HTML for structure, CSS for styling, and JavaScript for core logic and interactivity.

3. System Design

System Overview

The system follows a client-side architecture. The HTML provides the structure, the CSS defines the appearance and layout, and the JavaScript handles all the dynamic behavior and logic. The application is divided into two main sections: the Digital Clock and the Countdown Timer.

- Digital Clock: Uses the `Date` object in JavaScript to get the current time and the `setInterval` function to update the display every 1000 milliseconds (1 second).
- Countdown Timer: Manages a `totalSeconds` variable. User input is used to initialize this variable, and a `setInterval` function decreases it every second until it reaches zero.

Input & Output

Component	Type	Input/Output Description
Minutes Input (`#minutes`)	Input	User enters the desired minutes for the timer.
Seconds Input (`#seconds`)	Input	User enters the desired seconds for the timer.
Start/Pause/Resume Button (`#startBtn`)	Input/Output	Toggles the timer's running state. Button text changes based on state.

Reset Button (`#resetBtn`)	Input	Stops the timer and clears all values.
Current Time Display (`#clock`)	Output	Displays the current time in HH:MM:SS format.
Timer Display (`#timerDisplay`)	Output	Displays the remaining time in MM:SS format.
Notification Sound	Output	Plays a sound when the timer hits zero or every full minute during countdown.
Vibration Animation	Output	Briefly applies a CSS class to the container for a visual "vibrate" effect.

4. Implementation

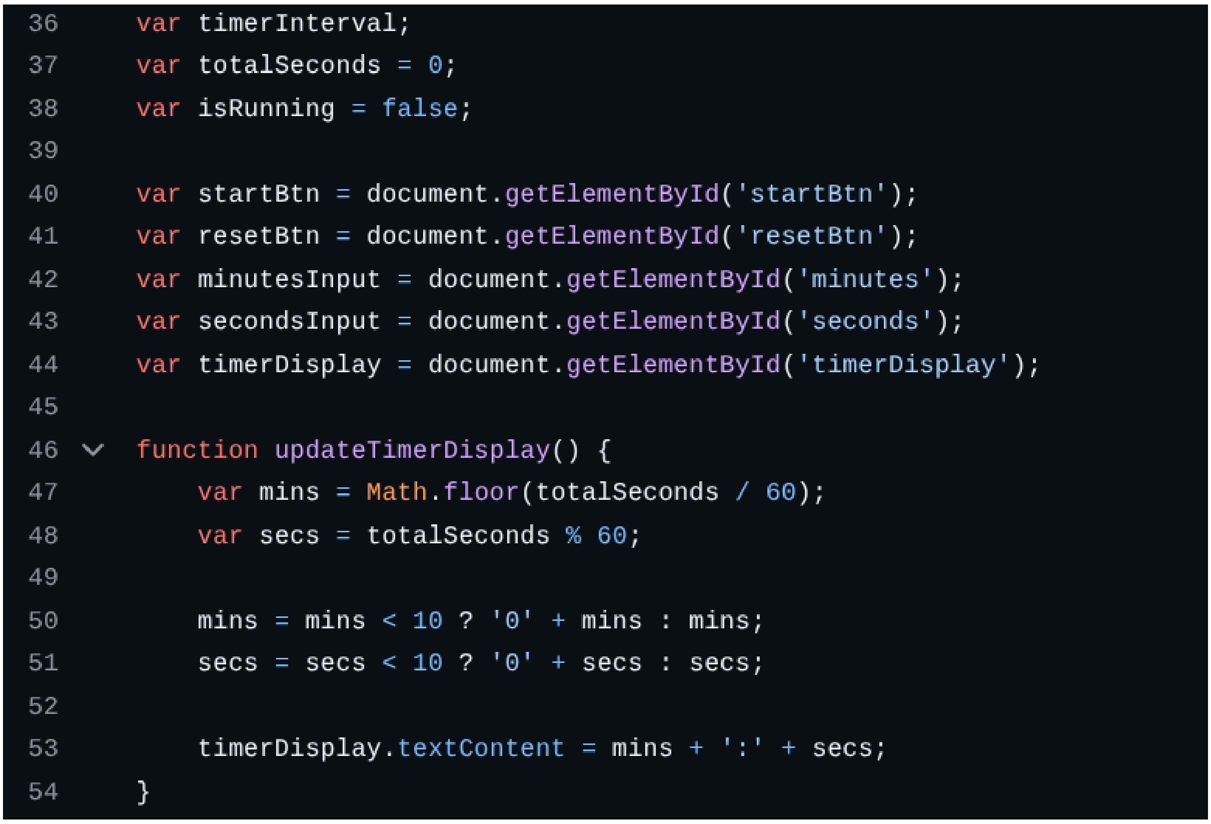
Language: HTML, CSS, JavaScript

Compiler/IDE: Visual Studio Code

Key Features

1. Real-Time Digital Clock: The `updateClock()` function runs every second via `setInterval(updateClock, 1000)` to show the system's current time.
2. Start/Pause/Resume Logic: The `startTimer()` function toggles the timer state using the `isRunning` boolean flag and manages `clearInterval()` and `setInterval()`. The button text dynamically updates to 'Pause', 'Resume', or 'Start'.
3. Time Input Handling: Initial timer duration is calculated from the Minutes and Seconds input fields: `totalSeconds = mins * 60 + secs;`.
4. Audio and Visual Notifications: The `playTimerSound()` function plays an audio file and adds the `vibrate-animation` CSS class to the main container, providing both audible and visual alerts. This triggers on timer completion and also every full minute remaining (e.g., 5:00, 4:00, etc.).

Code Snippet (JavaScript)



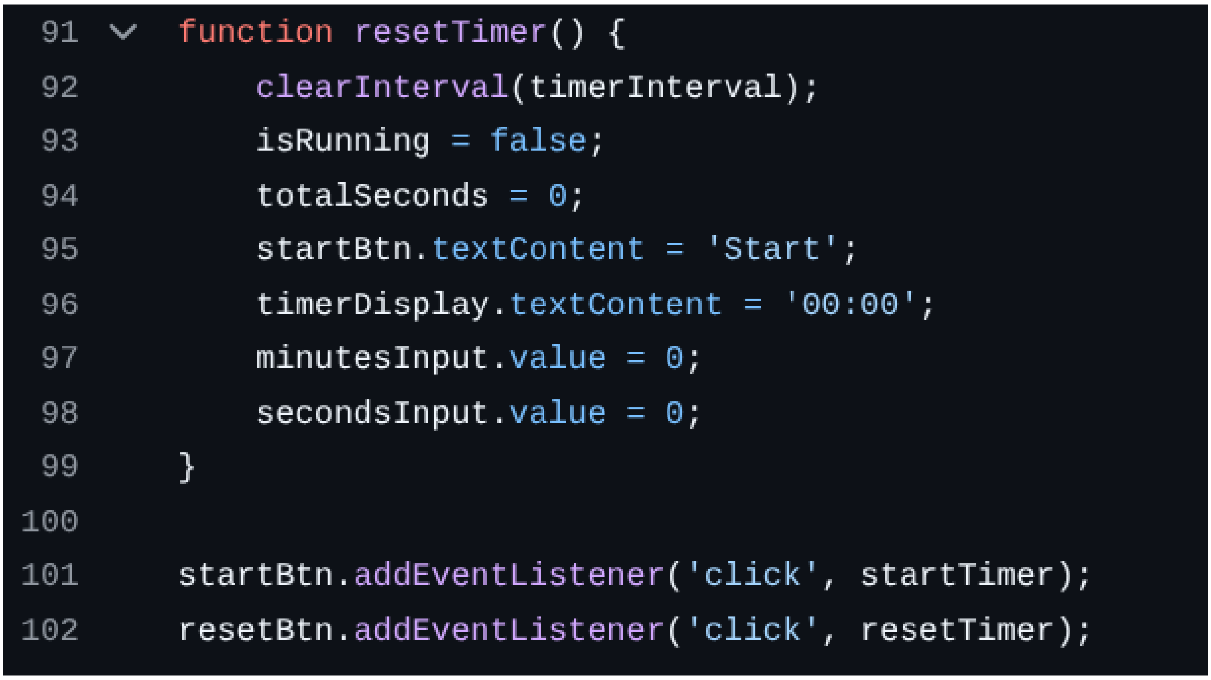
```javascript
var timerInterval;
var totalSeconds = 0;
var isRunning = false;

var startBtn = document.getElementById('startBtn');
var resetBtn = document.getElementById('resetBtn');
var minutesInput = document.getElementById('minutes');
var secondsInput = document.getElementById('seconds');
var timerDisplay = document.getElementById('timerDisplay');

function updateTimerDisplay() {
    var mins = Math.floor(totalSeconds / 60);
    var secs = totalSeconds % 60;

    mins = mins < 10 ? '0' + mins : mins;
    secs = secs < 10 ? '0' + secs : secs;

    timerDisplay.textContent = mins + ':' + secs;
}
```

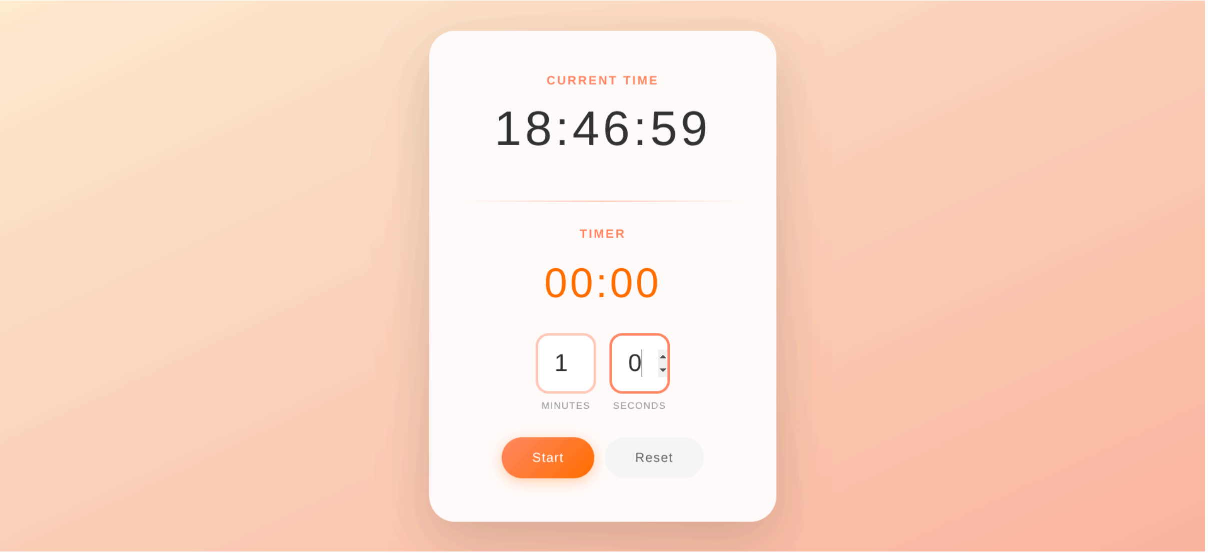


```
function resetTimer() {
    clearInterval(timerInterval);
    isRunning = false;
    totalSeconds = 0;
    startBtn.textContent = 'Start';
    timerDisplay.textContent = '00:00';
    minutesInput.value = 0;
    secondsInput.value = 0;
}

startBtn.addEventListener('click', startTimer);
resetBtn.addEventListener('click', resetTimer);
```

Sample Output

Timer set for one minute



CURRENT TIME

18:46:59

TIMER

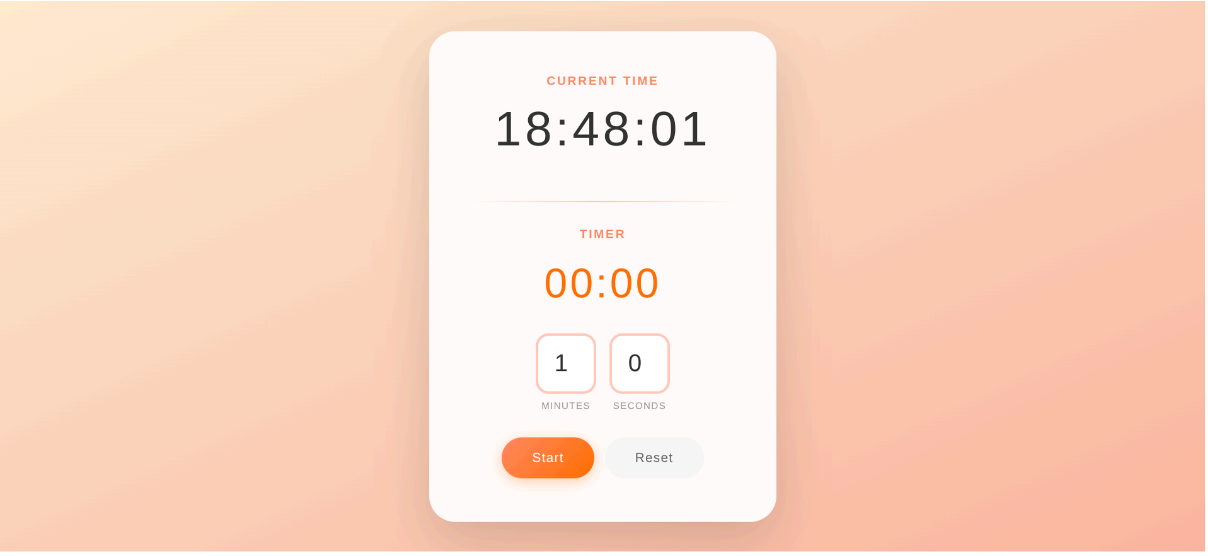
00:00

1 MINUTES

0 SECONDS

Output

A sound when timer is finished.



5.Testing & Results

Test Case	Expected Result	Actual Result
Digital Clock Update	Clock should update every second to show the current system time.	Pass
Start Timer	Timer should begin counting down from the set time, decrementing every second.	Pass
Pause/Resume	Timer should stop on 'Pause' and restart from the exact moment on 'Resume'.	Pass

Reset Timer	Timer should stop and reset the display and inputs to '00:00' and '0'.	Pass
Timer Completion	Timer should stop, reset the button to 'Start', and trigger the sound/vibration effect.	Pass
Minute Notification	Sound/vibration should trigger exactly when the minute count changes (e.g., from 02:00 to 01:59).	Pass

6. Conclusion, Limitations & References

Conclusion

The Digital Clock and Countdown Timer project was successfully developed using HTML, CSS, and JavaScript. It effectively implemented a real-time clock and a full-featured countdown timer with Start, Pause, Resume, and Reset functionalities, including audio and visual notifications. This project reinforced the understanding of JavaScript timing functions (`setInterval`) and building a responsive and user-friendly web interface.

Limitations

- No Persistence: The timer state and time inputs are lost if the browser tab is closed or the page is refreshed.
- Time Format: The digital clock only displays in the 24-hour format and does not offer an option to switch to 12-hour format (AM/PM).
- Alarm Only: The current design is a countdown timer; it does not feature a fixed-time alarm function (e.g., "alarm at 7:00 AM").

Future Enhancements

- Add local storage to save the timer's state, making it resilient to page refreshes.
- Implement a toggle button to switch the digital clock between 12-hour (AM/PM) and 24-hour formats.
- Include a feature for setting a one-time fixed-time alarm.
- Add a lap/split time feature to turn the application into a stopwatch.

References

- *W3Schools HTML, CSS, and JavaScript Tutorials*
- *Mozilla Developer Network (MDN) Web Docs -* `Date` *and* `setInterval`
- *Mixkit (for notification sound asset)*