A PROJECT REPORT

on

"COUNTDOWN TIMER"

Bachelor of Technology

in

COMPUTER SCIENCE ENGINEERING

Submitted by

M.KARTHIK	**237W1A0586**
M.GHAZANFER AHMED	**237W1A0594**
PRUDHVI SAI RAJU	**237W1A05B2**
M.ANJALI	**237W1A0592**

Under the guidance of

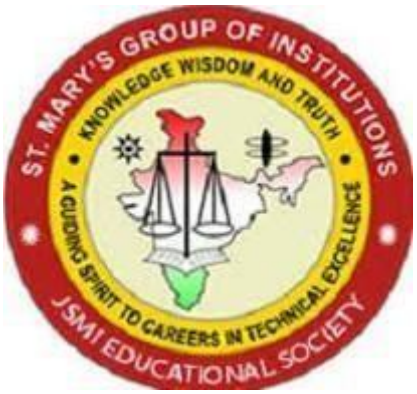
Mrs.A. NAGA LAXMI

(Asst.professor)

St.Mary's Integrated Campus,Hyderabad

(Affiliated to JNTU ,Hyderabad)

Deshmukhi(V),Pochampally(M),Yadadri.District-508115



JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY

HYDERABAD– 500072

(2024-2025)

UNDERTAKING

We declare that the work presented in this project titled **"COUNT DOWN TIMER"** is done by **M.KARTHIK** with team members **M.GHAZANFER AHMED, PRUDHVI SAI RAJU, M.ANJALI** We have not plagiarized or submitted the same work forthe award of any other examination. If this undertaking is found incorrect, we accept that our Certificates may be unconditionally withdrawn.

DATE:JUNE2025

PLACE:ST. Mary'sIntegratedCampus

CERTIFICATE

We Certify that the work contained in the project titled **"COUNT DOWN TIMER"** was done by **M.KARTHIK** with team members **M.GHAZANFER AHMED, PRUDHVI SAI RAJU, M.ANJALI** has been carried out under my supervision and this workhas not been submitted elsewhere for any other exams or projects.

(GUIDE NAME)
Mrs.A.Naga Laxmi
(ASSISTANTPROFESSOR)

(HEAD OF THE DEPARTMENT)
Dr.V.Samba Siva Rao
(PROFESSOR)

ACKNOWLEDGEMENTS

We would like tothank our DIRECTOR **Dr .SATYANARAYAN REDDY** sir, PRINCIPAL **DR.SWATIN GOUD** sir, and our HOD **Dr.V.SAMBA SIVA RAO** sir of St. Mary's College for giving me the opportunity to make this mini project. We own our sincere gratitude towards St. Mary's Integrated Campus.

Our heartfelt thanks to **Mrs.A.Naga Laxmi** mam for guiding us on this project.

We further thank all the staff members of St. Mary's Integrated Campus-Hyderabad,College.

We also express our deepest gratitude to our parents.

Finally, we would like to wind up by paying our heartfelt thanks to all our near and dear ones.

``

1. M.KARTHIK -CSE ROLLNO-237W1A0586
2. M.GHAZANFER AHMED -CSE ROLLNO-237W1A0594
3. PRUDHVI SAI RAJU -CSE ROLLNO- 237W1A05B2
4. M.ANJALI -CSE ROLLNO- 237W1A0592

INDEX

S.NO	TOPIC	Pg.NO
1	ABSTRACT OF THE PROJECT	06
2	INTRODUCTION OF THE PROJECT	07
3	PROBLEM STATEMENT	08
4	OBJECTIVE	09
5	SOFTWARE REQUIREMENTS	10
6	SOFTWARE ENVIRONMENT	10
7	CODE OF THE PROJECT	11
8	SAMPLE OUTPUT OF THE PROJECT	21
9	CONCLUSION OF THE PROJECT	22

ABSTRACT

COUNTDOWN TIMER

The **Countdown Timer** is a simple yet practical software project designed to count down from a specified time duration to zero, providing a clear and user-friendly display throughout the countdown process. This project demonstrates the implementation of real-time timer functionality using programming techniques, offering a visual representation of minutes and seconds that updates continuously.

Key features of the Countdown Timer include customizable time duration input, real-time countdown display, alarm or notification upon completion, and a graphical user interface (GUI) for improved user interaction. The system ensures precise timing by utilizing the system clock and updating the display every second for accuracy and reliability. The interface is designed to be intuitive, making it easy for users to set, start, pause, and reset the timer.

The implementation of the Countdown Timer offers educational benefits such as a deeper understanding of time-based operations, GUI development, and event-driven programming. While relatively simple, the project serves as a strong foundation for learning core programming concepts and can be extended to more advanced time-management or scheduling applications. It is especially beneficial for beginners seeking hands-on experience in real-time application development.

INTRODUCTION

The **Countdown Timer** project, built as a web-based application using HTML, CSS, and JavaScript, provides a practical and efficient solution for measuring time intervals through a graphical user interface. This project focuses on creating a responsive and visually appealing system that allows users to set a specific time duration, which then counts down to zero using modern web technologies.

The web-based application allows users to interact with the system through a browser, making it lightweight and easy to use without the need for any external software installation. Key features of this project include real-time countdown functionality, smooth user interface design, and a responsive layout that adapts to different devices. Utilizing JavaScript's built-in timing functions and DOM manipulation, the application can efficiently manage and display the countdown in real-time.

For instance, the project uses HTML to structure the timer elements, CSS for designing an aesthetically pleasing layout, and JavaScript to handle countdown logic and real-time updates. The application may also include features like start, pause, reset buttons, alert notifications, and digital styling, ensuring an engaging and functional user experience.

This web-based application is ideal for beginners and learners looking for a cost-effective, easy-to-implement solution to understand the integration of HTML, CSS, and JavaScript. It demonstrates how a simple yet effective tool can significantly enhance the understanding of real-time web applications, UI development, and event-driven programming.

.

PROBLEM STATEMENT

The use of countdown timers in digital platforms often faces several challenges, including inaccurate time tracking, lack of real-time responsiveness, and poorly designed user interfaces. Traditional methods, such as manual timers or non-interactive countdown tools, fail to provide customizable features or responsive layouts suited for modern web applications. These limitations can result in reduced usability, outdated visual presentation, and inefficient time management in web-based environments.

OBJECTIVE

To address these challenges, we propose developing a web-based **Countdown Timer** application using HTML, CSS, and JavaScript. This system aims to:

● **Enable Real-Time Countdown Functionality:** Provide accurate and continuously updating countdown behavior using JavaScript's timing functions.
● **Deliver a Responsive User Interface:** Design an intuitive and visually appealing layout that adapts across various screen sizes and devices.
● **Allow Custom Countdown Durations:** Enable users to input specific time durations for countdowns, offering flexibility for different use cases.
● **Simplify Web-Based Timer Integration:** Offer an easy-to-implement, browser-based solution that can be embedded into other web projects or used as a standalone countdown tool.

By creating a lightweight, client-side application, this project aims to deliver a simple, efficient, and visually engaging solution for time-based interactions in web environments, enhancing both usability and user experience.

SOFTWARE REQUIREMENTS

● **HTML5**:The core markup language used for structuring the digital clock interface and displaying time elements on the webpage.
● **CSS3**:Used to style and enhance the visual appearance of the clock, Including layout,colors,fonts,and responsive design.
● **JavaScript (ES6+)**: The main scripting language used to implement real-timetime functionality,update the clock every second ,and handle time format logic.
● **Web Browser**: Acts as the platform for running and displaying the application, requiring no additional installations or dependencies.

SOFTWARE ENVIRONMENT

● ● **Operating System:** The application should be compatible with all major operating systems, including Windows, macOS, Linux, and mobile platforms through modern web browsers.
● **Code Editor/IDE:** Recommended editors include Visual Studio Code, Sublime Text, or any IDE that supports front-end web development using HTML, CSS, and JavaScript.
● **Web Browser:** A modern browser such as Google Chrome, Mozilla Firefox, Microsoft Edge, or Safari is required to run and view the countdown timer.
● **Version Control System (Git):** For tracking code changes, collaborating with others, and maintaining the project's source code efficiently..

COMPLETE CODE

HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <link rel="stylesheet" href="style.css">
   <title>countdown timer</title>
</head>
<body>
   <!-- countdown form-->
   <div class="form-area">

      <form id="counter-form">
         <div>
            <label>Enter Countdown Title:</label>
            <input type="text" name="title" id="counter-title-input" placeholder="what's you're
counting to ?">

         </div>
         <div>
            <label>Pick a Date</label>
            <input type="date" name="date-picker" id="counter-date">

         </div>
         <button type="submit" id="counterSubmit">Submit</button>
      </form>
   </div>


   <!--MAIN COUNTER-->
   <div class="counter" id="counter" hidden>

      <h1 id="counter-title">Counter Title</h1>
      <ul>
         <li><span>3</span>Days</li>
         <li><span>6</span>Hours</li>
         <li><span>12</span>Minutes</li>
         <li><span>23</span>Seconds</li>
      </ul>
      <button type="reset" id="counter-reset">Reset</button>
   </div>

   <!--complete screen-->

   <div class="complete" id="complete">
      <h1 class="complete-title">Counter Completed..!</h1>
      <h1 id="complete-info">Countdown finished on 26-05-2025</h1>
      <button id="complete-button">New Countdown</button>
   </div>
   <script src="script.js"></script>
</body>
</html>
```

CSS:

```
body{
    background-color: #2A7B9B;
    background-image: linear-gradient(90deg, rgba(42, 123, 155, 1) 0%, rgba(87, 199,
133, 1) 50%, rgba(237, 221, 83, 1) 100%);
    font-family: Arial, Helvetica, sans-serif;
}


.form-area, .counter, .complete{
    padding: 20px;
    border-radius: 5px;
    box-shadow: #1c3446;
    background-color: white;
    width: fit-content;
    margin-left: auto;
    margin-right: auto;
    margin-top: 10%;

}


.form-area input{
    display: block;
    margin: 15px;
    padding: 10px;
    margin-left: 0px;
    margin-right: 0px;
    width: 300px;
}
.form-area{
    font-size: 14px;
    font-weight:bold ;
}

button{
    background-color: rgb(0, 110, 255);
    width: 100%;
    padding: 15px;
    border: none;
    color: white;
    font-weight: bold;
    font-size: 16px;
    border-radius: 5px;
}
button:hover{
    background-color: rgb(0, 89, 172);
    cursor:pointer ;
}

.counter h1{
    text-align: center;
}
```

```
li{
    display: inline-block;
    font-size: 30px;
    list-style-type: none;
    padding: 10px;
    text-transform: uppercase;
    color: rgb(27, 27, 27);
}

li span{
    display: block;
    font-size: 80px;
    text-align: center;
}


.complete-title{
    animation: complete 4s infinite;
    text-align: center;
}
@keyframes complete{
    0%{
        color: red;

    }
    25%{
        color: yellow;
        transform: rotate(5deg);
    }
    50%{
        color: green;
        transform:scale(1.2);
    }
    75%{
        color: blue;
        transform: rotate(-5deg);
    }
    100%{
        color: red;
        text-replace: rotate(0deg);
    }
}
```

JAVASCRIPT:

```
f

const counterFormArea =
    document.querySelector ('.form-
    area');

const counterForm =
    document.getElementById
    ('counter-form');

const counterE1 =
    document.getElementById
    ('counter');

const counterTitleE1 =
    document.getElementById
    ('counter-title');

const timeElements =
    document.querySelectorAll
    ('span');

const counterResetBtn =
    document.getElementById
    ('counter-reset');




const complete =
    document.getElementById
    ('complete');

const completeInfo =
    document.getElementById
    ('complete-info');



const completeBtn =
    document.getElementById
    ('complete-button');
```

```
const datePicker =
    document.getElementById
    ('counter-date');


let countdownValue = Date;

let countdownActive;


const second = 1000;

const minute = second * 60;

const hour = minute * 60;

const day = hour * 24;


let title = '';

let date = '';


let today = new
    Date().toISOString().split ('T')[0];


console.log (today);


datePicker .setAttribute('min',today);



function updateDom (){
```

```
countdownActive = setInterval (()
 => {

   let now = new Date().getTime();

   let distance = countdownValue -
now;

   // console.log (distance);

   const days =
Math.floor(distance/day);

   const hours = Math.floor(distance
% day/hour);

   const minutes =
Math.floor(distance%hour/minute);

   const seconds =
Math.floor(distance%minute/secon
d);









   if (distance < 0) {

      counterE1.hidden = true;

      counterFormArea.hidden =
true;

      complete.hidden = false;

      clearInterval
(countdownActive);

      completeInfo.textContent
=`${title} is finished on ${date}`;
```

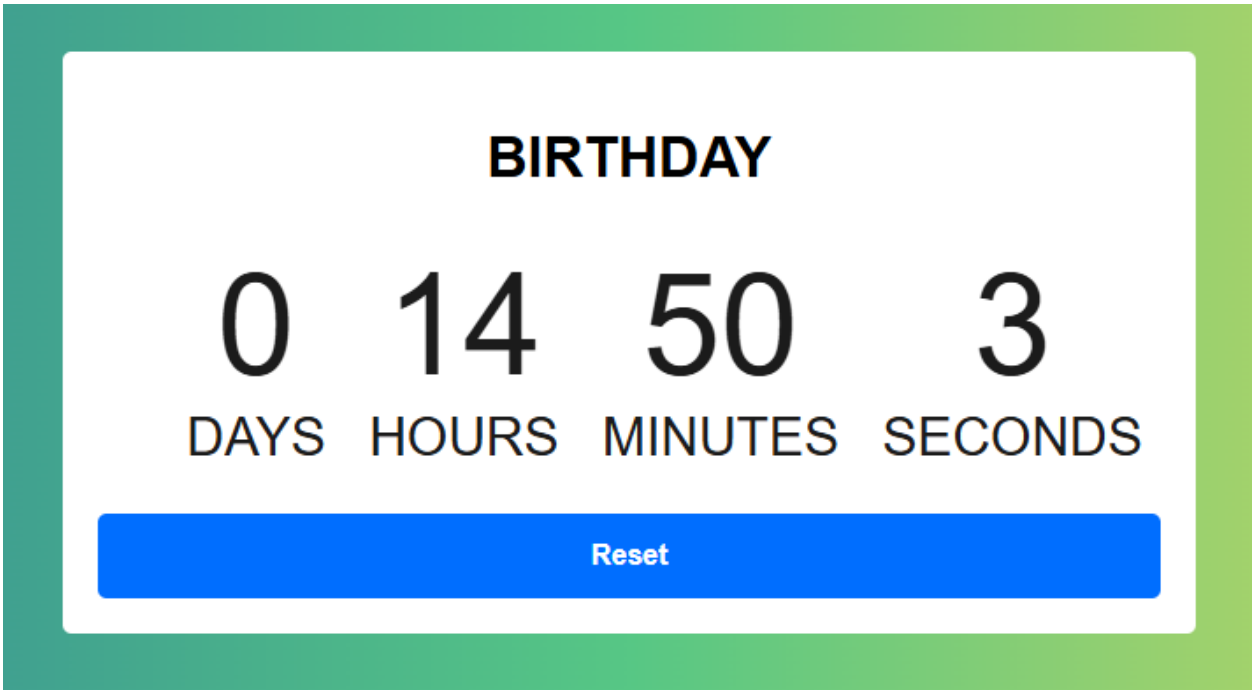
```
      } else {

        timeElements[0].textContent =
      days;

        timeElements[1].textContent =
      hours;

        timeElements[2].textContent =
      minutes;

        timeElements[3].textContent =
      seconds;

        counterTitleE1.textContent =
      title;

        counterFormArea.hidden = true;

        counterE1.hidden = false;



      }

    }, 1000);

}









function updateCountdown (e) {

    e.preventDefault ();

    title = e.srcElement[0].value;
```

```
    date = e.srcElement[1].value;


    const savedCountdown = {
        title:title,
        date:date,
    }


     localStorage.setItem('countdown',J
     SON.stringify(savedCountdown));


    console.log (title, date);
    if(date === '') {
        alert ('Please enter a date!');
    } else {
        countdownValue = new Date
      (date).getTime ();
        console.log (countdownValue);
        updateDom ();

    }
}
```

```
function reset(){
    counterE1.hidden = true;
    complete.hidden = true;
    clearInterval(countdownActive);
    title = '';
    date = '';
    counterFormArea.hidden = false;
}


function restoreCountdown(){

    if(localStorage.getItem('countdown
    ')){
      counterFormArea.hidden = true;
      let countdownData =
    JSON.parse(localStorage.getItem('
    countdown'));
      title = countdownData.title;
      date = countdownData.date;
      countdownValue = new Data
    (date).getTime ();
      updateDom();

   }
```

```
}



counterForm.addEventListener('submi
    t', updateCountdown);

counterResetBtn.addEventListener('cli
    ck',reset);

completeBtn.addEventListener('click',r
    eset);



restoreCountdown ();14
```

SAMPLEOUTPUT



CONCLUSION

The **Countdown Timer** offers a simple yet effective solution for tracking time intervals in a visually appealing and responsive manner. This web-based application, developed using HTML, CSS, and JavaScript, allows users to set and monitor countdowns accurately and continuously through a clean and interactive interface. By providing a streamlined and user-friendly design, the system enhances usability, supports customizable time durations, and ensures compatibility across various devices and platforms. Its lightweight and easily deployable structure makes the Countdown Timer a valuable tool for learning, integration into other web projects, or use as a standalone time management component in modern web environments.

.