

**SIX WEEK SUMMER TRAINING**

**REPORT**

on

**JavaScript for Web Development**

Submitted by

Name: Pavuluri Chaitanya

Reg.No:12103286

In the partial fulfilment for the requirements of the award of the degree of

**BACHELOR OF TECHNOLOGY (B. TECH)**

Under the Guidance of

**Akshay Mittal**

Industry Coordinator Boord Infinity

**School Of Computer Science & Engineering**

**Lovely Professional University, Phagwara**

**Student Declaration**

I hereby declare that I have completed my six weeks summer training at Board infinity from 01-06-23 to 12-07-23 under the guidance of Akshay Mittal. I have declare that I have worked with full dedication during these six weeks of training and my learning outcomes fulfil the requirements of training for the award of degree of B.tech Computer Science Engineering (Full Stack Web-Development), Lovely Professional University, Phagwara.

Pavuluri Chaitanya

Registration no:12112110

Dated: 24-08-2023

**Summer training Certificate:**

****

# Introduction:

I am writing this report on my experience in the last 6-week summer training internship on JavaScript for web development. During this time, I have learned the fundamentals of JavaScript, including the basics of JavaScript syntax and semantics, variables, data types, and operators, functions, objects and arrays, events and event handling, DOM manipulation.

I have also applied these concepts to build a real-world web application and also build three different project which is very useful in the real life. This JavaScript project consists of three main components: a digital clock, a random quote generator, and an ID card generator. The project is mainly focused on the JavaScript part, with HTML and CSS used for the front-end.

1.Digital Clock: The digital clock uses JavaScript to update the time every second, With Start and Stop functionality buttons

2.Random Quote Generator: The random quote generator uses JavaScript to fetch a random quote from a list of quotes. The quotes are stored in a JavaScript part. The quote is then styled using CSS to make it look good on the page.

3.ID Card Generator: The ID card generator uses JavaScript to generate a ID card. The ID card contains the Student Details with student name, college name, location and student photograph. The ID card is styled using CSS.

In additional, I have also worked on a number of other projects. These projects have helped me to understand concepts of JavaScript and to gain experience in building web applications.

I am grateful for the opportunity to have participated in this training internship. I have learned a great deal and I am confident that I can use these skills to become a successful web developer.

# Technology Learnt:

In this six-week summer training summer internship course I have learnt so many Technologies, like html, CSS and JavaScript etc. particularly in JavaScript I have explored more

* **JavaScript syntax and semantics:** This is the foundation of JavaScript, and it is essential for understanding how the language works.
* **Variables, data types, and operators**: These are the building blocks of JavaScript, and it will need to know how to use them to store data and perform calculations.
* **Objects and arrays:** Objects and arrays are data structures that allow it to store and organize data in a more efficient way.
* **Events and event handling:** Events are notifications that are triggered by user interaction or other events. Event handling allows it to respond to these events in a desired way.
* **DOM manipulation:** The DOM (Document Object Model) is the underlying structure of a web page. DOM manipulation allows it to change the content, style, or behaviour of a web page dynamically.

# Introduction of the company

# Board Infinity:

Board Infinity is an online learning platform that offers courses in technology, data science, and management. They offer self-paced, live cohort, offline, and 1:1 courses taught by top industry experts. They also offer placement preparation, mock interviews, career services, and job assurance.

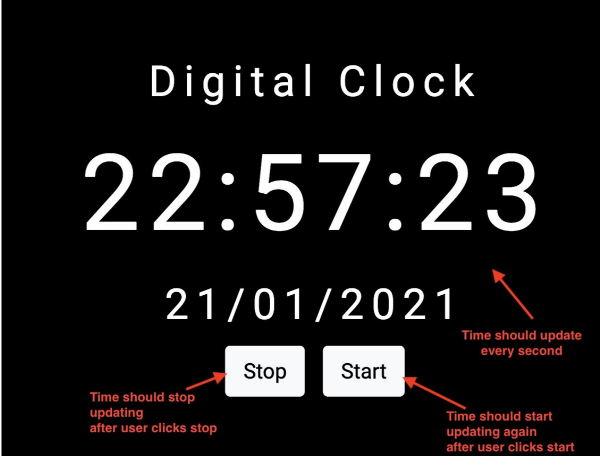
Board infinity appears to be a promising platform that seeks to address the challenges and uncertainties often associated with career development. From my perspective, it seems that the company understands the unique struggles and aspirations of students and jobseekers. Their emphasis on personalized learning paths and access to career coaches resonates with me as it indicates a commitment to providing tailored support rather than a one-size-fits-all approach.

# Profile of the problem:

Build a digital clock which should display current time and update the time after every second.

When the user clicks `Stop` Button the clock should stop updating the time. When the user clicks on the `Start` button later the clock should start updating the current

time again as shown below: -



Learnings:

How to make use of set Interval and clear Interval method in JavaScript.

# Problem Analysis of JavaScript

Scope: JavaScript has a dynamic scope, which means that the scope of a variable can change depending on where it is declared. This can make it difficult to track down bugs and can lead:

Type coercion: JavaScript automatically converts values of different types when they are combined. This can lead to unexpected results and can make it difficult to write robust code.

Performance: JavaScript can be slow, especially when it is used to perform complex calculations or animations. This can be a problem for applications that need to be responsive.

**Problem analysis for Problem 1:**

Problem:

* The clock should display the current time in a human-readable format, such as hours, minutes, and seconds.
* The clock should update the time every second.
* The clock should have a stop button that pauses the time updates.
* The clock should have a start button that resumes the time updates.

Solution:

The following are the steps involved in building a digital clock that meets the above requirements:

* Get the current time from a time source, such as the operating system or a hardware clock.
* Convert the current time to a human-readable format.
* Display the current time on a display device, such as an LED display or a LCD screen.
* Create a timer that fires every second.
* In the timer's event handler, update the current time and display it on the display device.
* Add a stop button to the clock. When the stop button is clicked, disable the timer.
* Add a start button to the clock. When the start button is clicked, enable the timer.

# Feasibility Analysis:

Feasibility Analysis of problem :

The feasibility of building a digital clock that displays the current time and updates the time after every second, and stops updating the time when the user clicks the Stop button and starts updating the time again when the user clicks the Start button depends on the following factors:

* Technical feasibility: The clock can be built using a variety of technologies, including electronic circuits, software, and mechanical devices. The specific technology that is used will depend on the desired features and capabilities of the clock.
* Financial feasibility: The cost of building the clock will depend on the materials and components that are used, as well as the complexity of the design. The clock can be built for a relatively low cost if simple materials and components are used.
* Market feasibility: There is a market for digital clocks, so the clock is likely to be successful if it is well-designed and meets the needs of the target market.
* Operational feasibility: The clock must be able to operate reliably and consistently. The design of the clock must take into account factors such as the environment in which it will be used, the frequency of use, and the expected lifespan.

Overall, the feasibility of building a digital clock that displays the current time and updates the time after every second, and stops updating the time when the user clicks the Stop button and starts updating the time again when the user clicks the Start button is high. The clock can be built using a variety of technologies, and there is a market for digital clocks. The key to success is to design a clock that is well-made and meets the needs of the target market.

# Software Requirement Analysis

software requirement analysis for problem 1:

software requirements for a digital clock that displays the current time and updates the time after every second, and stops updating the time when the user clicks the Stop button and starts updating the time again when the user clicks the Start button:

Functional requirements:

* The clock must display the current time in a human-readable format, such as hh:mm:ss.
* The clock must update the time after every second.
* The clock must have a Stop button that stops the clock from updating the time.
* The clock must have a Start button that starts the clock updating the time again.

Non-functional requirements:

* The clock must be accurate to within one second.
* The clock must be user-friendly.
* The clock must be able to run on a variety of platforms, such as Windows, macOS, and Linux.

software requirement analysis for problem 2:

software requirements for a random quote generator that displays a greeting statement depending on the time of day, and shows a different quote after clicking the New Quote button with a new theme in the background:

Functional requirements:

* The generator must display a greeting statement in the title section depending on the time of day.
* The generator must display a different quote after clicking the New Quote button.
* The generator must randomly choose a color from a collection of at least 10 colors and apply it to the background of the page, the button, and the quote text.

Non-functional requirements:

* The generator must be user-friendly.
* The generator must be able to run on a variety of platforms, such as Windows, macOS, and Linux.

# Main Problem:

**Problem 1: Digital clock**

Digital clock displays the current time and date Where time is in 24 Hour format and date is in mm/dd/yd.

In this code I have usen internal CSS and JavaScript.

**Code:**

<!DOCTYPE html>

<html>

<head>

<title>Digital Clock</title>

<style>

    body {

        font-family: Arial, sans-serif;

        text-align: center;

        margin-top: 50px;

        background-color: black;

        background-image: url('https://images.pexels.com/photos/41949/earth-earth-at-night-night-lights-41949.jpeg?cs=srgb&dl=pexels-pixabay-41949.jpg&fm=jpg');

        }

        h1 {

        font-size: 24px;

        color: white;

        }

        #clock {

        font-size: 80px;

        margin-top: 20px;

        color: white;

        }

        #date {

        font-size: 24px;

        margin-top: 10px;

        color: white;

        }

        button {

        font-size: 16px;

        padding: 10px 20px;

        margin-top: 20px;

    }

</style>

</head>

<body>

<h1>Digital Clock</h1>

<div id="clock"></div>

<div id="date"></div>

<button id="startBtn">Start</button>

<button id="stopBtn">Stop</button>

<script >

    var clockElement = document.getElementById('clock');

var dateElement = document.getElementById('date');

var isRunning = false;

var timerId;

document.getElementById('startBtn').addEventListener('click', function() {

if (!isRunning) {

isRunning = true;

timerId = setInterval(updateTime, 1000);

}

});

document.getElementById('stopBtn').addEventListener('click', function() {

if (isRunning) {

isRunning = false;

clearInterval(timerId);

}

});

function updateTime() {

var now = new Date();

var options = { hour: '2-digit', minute: '2-digit', second: '2-digit',

hour12: false };

var time = now.toLocaleTimeString([], options);

var date = now.toLocaleDateString();

clockElement.textContent = time;

dateElement.textContent = date;

}

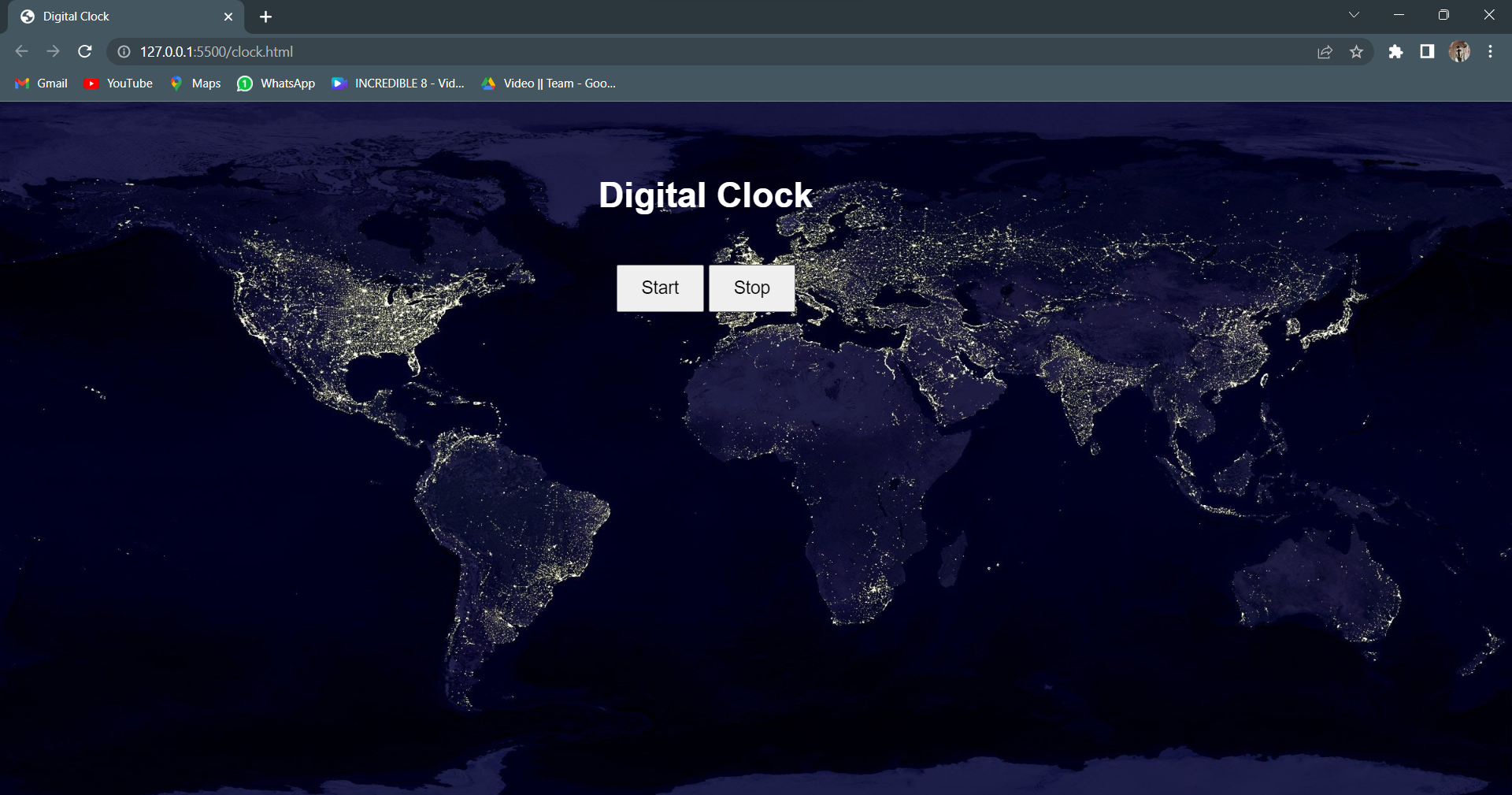
</script>

</body>

</html>

**Output:**

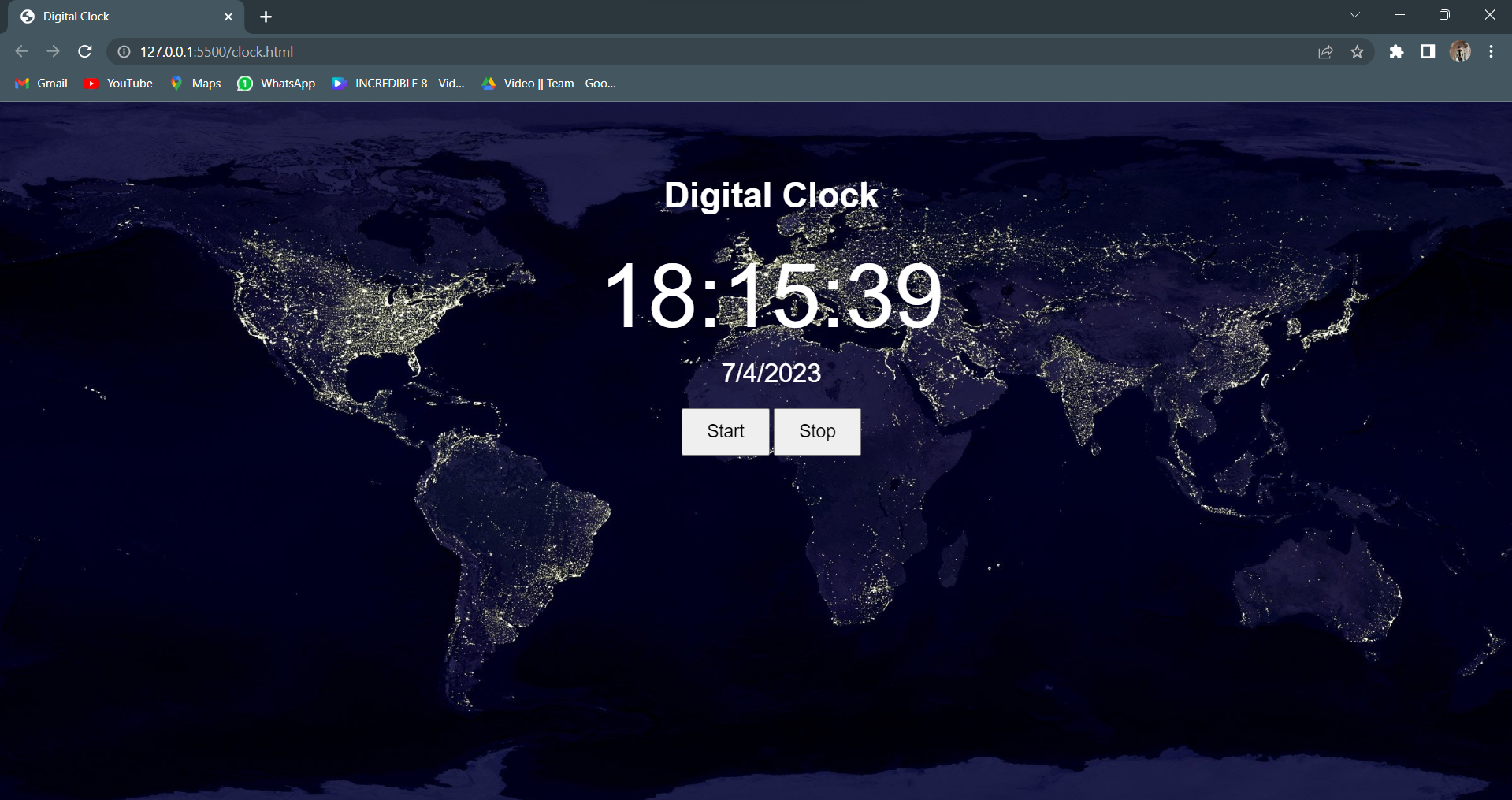
When ever we click this button current time will be shown

****

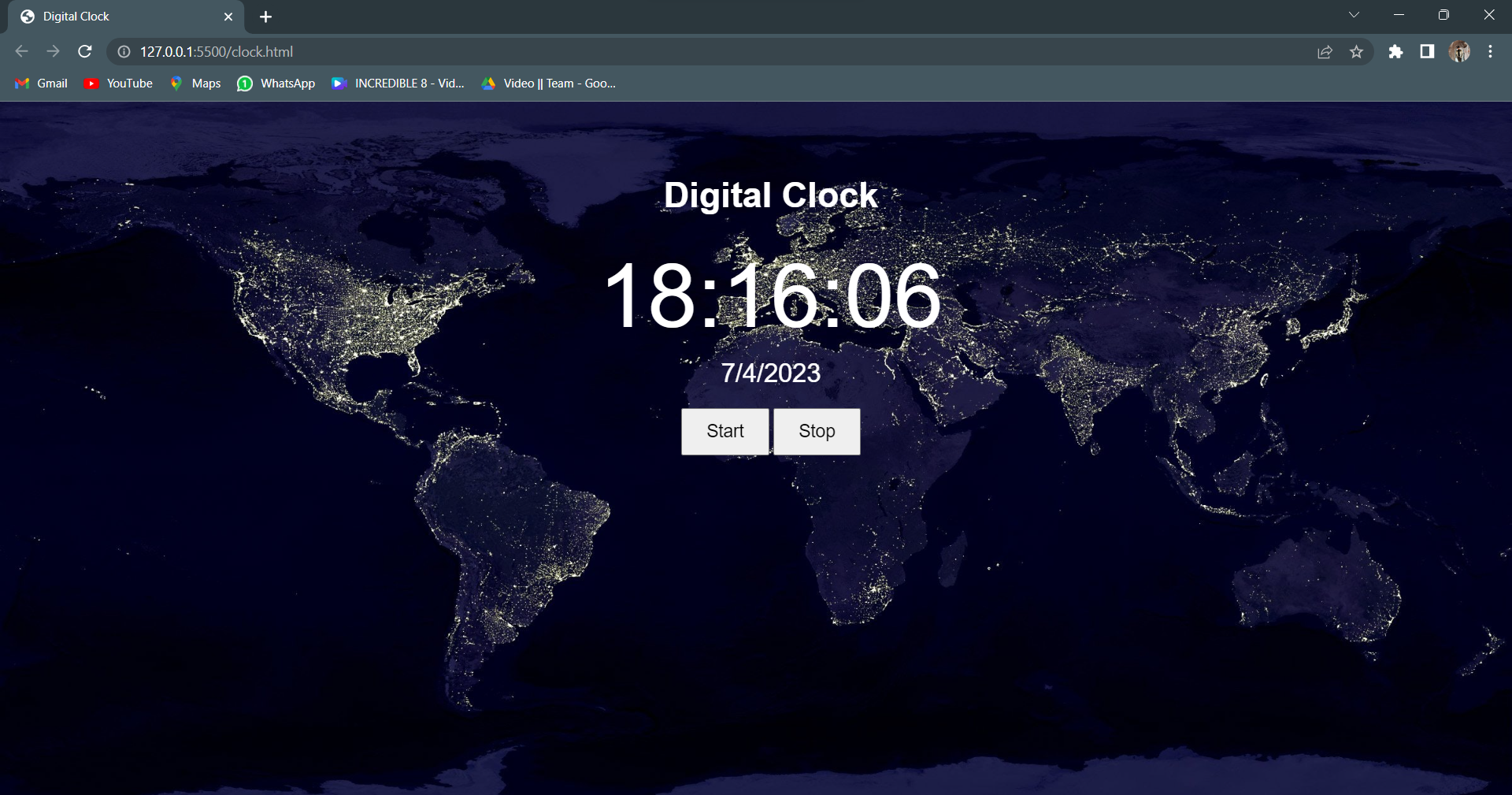
Background image from

Online

Time will be stopped until we click on start button and after clicking start button the current time will be shown



After clicking on the start button this output was shown



# Learning outcome from training/ technology learnt:

* Understand the basic concept of JavaScript syntax and semantics.
* Be able to write and use JavaScript variables, expressions, and statements.
* Understand the concept of objects and how to use them in JavaScript.
* Be able to write functions and use them to modularize code.
* Understand the basics of JavaScript control flow, including loops and conditional statements.
* Be able to use JavaScript to manipulate the DOM (Document Object Model).
* Be able to use JavaScript to create interactive web pages.
* Be familiar with common JavaScript libraries and frameworks.
* Be able to debug JavaScript code.
* Be able to create interactive web pages. JavaScript can be used to add interactivity to web pages, such as making buttons that change color when they are clicked, or displaying a message when the user hovers over an image.

# 

# Project Legacy

**Technical lessons:**

JavaScript is a versatile language that can be used for a variety of tasks, such as creating interactive web pages, developing games, and controlling IoT devices.

JavaScript is a client-side language, which means that it is executed on the user's browser. This makes it ideal for creating dynamic and interactive web pages.

JavaScript is a powerful language that can be used to create complex applications. However, it is also a complex language, and it takes time and practice to master.

There are many different JavaScript frameworks and libraries available, each with its own strengths and weaknesses. It is important to choose the right framework or library for the task at hand.

**Managerial lessons:**

It is important to have a good understanding of the software development process, including the different phases of development, such as requirements gathering, design, implementation, testing, and deployment.

It is also important to be able to work effectively as part of a team, and to communicate effectively with both technical and non-technical stakeholders.

It is important to be able to manage your time effectively, and to be able to work under pressure.

It is also important to be able to learn new things quickly, and to be adaptable to change.

# Bibliography

JavaScript: The Definitive Guide by David Flanagan is a comprehensive book that covers all aspects of JavaScript, from the basics to advanced topics.

JavaScript Testing Patterns by Michael C. Davis is a book that covers testing JavaScript code.

JavaScript: The Definitive Guide

**Reference links**

Board infinity: <https://www.boardinfinity.com/>

Mozilla Developer Network (MDN): <https://developer.mozilla.org/en-US/docs/Web/JavaScript>

JavaScript Tutorial: <https://www.tutorialspoint.com/javascript/>

W3Schools: <https://www.w3schools.com/js/>

JavaScript Forum: <https://forum.freecodecamp.org/c/javascript>