Front End Engineering-II

Project Report

Semester-IV (Batch-2022)

DIGITAL AND ANALOG CLOCK

Supervised By:

Dr. Raveesh Samkaria

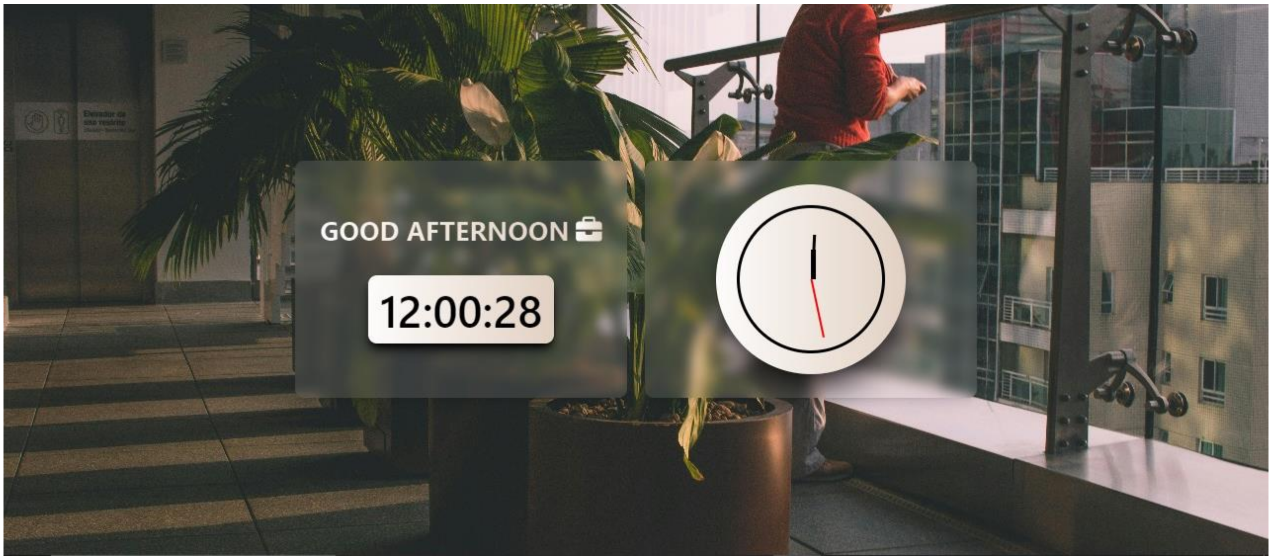
Submitted By:

Rudrakshi, 2210990747 (G-12)

Department of Computer Science and Engineering
Chitkara University Institute of Engineering &Technology,
Chitkara University, Punjab

Abstract

This project develops a dual clock system with digital and analog clock components using HTML, CSS with the Tailwind framework, and JavaScript. The digital clock displays the current time, updated every second via JavaScript. The analog clock features hour, minute, and second hands dynamically positioned based on the current time. Tailwind CSS provides a modern, responsive design across devices. The clocks can integrate into web apps or function as standalone components. The project offers practical experience with web technologies, time manipulation, DOM manipulation, and building interactive, visually appealing components.



INDEX

1. Introduction

In the digital age, clocks have become an integral part of our daily lives, serving as a constant reminder of the passage of time. While digital clocks have gained widespread popularity due to their precision and ease of reading, analog clocks continue to hold a special place in our hearts, often associated with a sense of nostalgia and traditional timekeeping. This project aims to create a comprehensive solution that combines the best of both worlds, offering users a seamless experience with both digital and analog clock representations.

1.1 Background:

The concept of telling time has evolved significantly over the centuries, from sundials and water clocks to mechanical and electronic timepieces. With the advent of modern technology, digital clocks have become ubiquitous, offering a clear and precise display of time. However, analog clocks, with their elegant design and intricate movements, have remained a beloved choice for many individuals, adding a touch of sophistication and artistry to timekeeping.

1.2 Objectives:

The core goals of this project involve developing a functional and visually appealing implementation of both digital and analog clocks using web technologies. It leverages HTML for structuring the clock components, CSS with the Tailwind framework for styling, and JavaScript for dynamic updates and interactivity. Furthermore, the project aims to ensure the clocks are responsive and display correctly across various devices and screen sizes and create a modular and reusable codebase that can be easily integrated into web applications or used as standalone components.

1.3 Significance:

The project holds significant value as it provides a practical exercise in building interactive and visually appealing web components, combining digital and analog time representations. It offers hands-on experience with HTML, CSS (including Tailwind framework), and JavaScript, reinforcing essential web development skills. Additionally, it explores concepts like time manipulation, DOM manipulation, and event handling

2. Problem Statement

In the digital age, while digital clocks have become ubiquitous due to their precision and ease of reading, there is still a demand for analog clocks, which offer a traditional and aesthetically pleasing way of representing time. However, developing a solution that seamlessly integrates both digital and analog clocks, while ensuring responsiveness across multiple devices and providing additional features, can be a challenging task.

3. Software Requirements

I. Integrated Development Environment (IDE):

- Visual Studio Code (VS Code) will be utilized as the primary code editor and project management tool for its robust features, extensive extension ecosystem, and efficient workflow.

II. Technology Requirements:

- HTML5 (Hypertext Markup Language) will be employed for structuring the web application and creating the necessary elements for the clock components and user interface.
- CSS3 (Cascading Style Sheets) will be used for styling the application, ensuring a visually appealing and coherent design across different components and screen sizes.
- JavaScript (ES6+) will be the programming language of choice for implementing interactive features, dynamic updates, time calculations, and handling user interactions within the application.

III. Utility-first CSS Framework:

- Tailwind CSS, a highly customizable and utility-first CSS framework, will be leveraged to streamline the styling process and achieve a modern, responsive design with minimal effort.

IV. Version Control and Collaboration:

- Git, a distributed version control system, will be utilized for tracking changes in the project codebase, enabling efficient collaboration, and facilitating code reviews and merges.
- The project repository will be hosted on GitHub, a popular platform for code hosting, collaboration, and issue tracking.

V. Browser Compatibility and Performance:

- The application will be developed with a focus on cross-browser compatibility, ensuring it functions correctly and consistently across the latest versions of popular web browsers.
- Performance optimization techniques will be implemented to ensure smooth and efficient rendering, minimizing resource consumption and providing an optimal user experience.

4. Proposed Design

User Interface:

- The application will feature a responsive and visually appealing layout using Bootstrap 5, ensuring a consistent experience across various devices and screen sizes.
- A card-based design will be employed to organize the different components of the application, such as the digital clock, analog clock, and time format/date display options, in an intuitive and user-friendly manner.

Development Technologies:

- HTML5 semantic elements will be utilized for structuring the application, ensuring proper semantics and accessibility.
- CSS will be responsible for styling the application, creating a visually appealing and cohesive design.
- JavaScript will be used for implementing dynamic UI updates, handling user interactions, and managing the clock logic and time calculations.

User Experience:

- Real-time feedback and interactive elements, such as smooth animations and transitions, will be implemented to create an engaging and responsive user interface.
- Cross-browser compatibility will be a priority, ensuring the application functions consistently across the latest versions of popular web browsers.
- Responsiveness will be a key focus, with the application designed to adapt seamlessly to different devices and screen sizes, providing an optimal user experience on both desktop and mobile platforms.

Testing and Quality Assurance:

- Comprehensive testing, including manual and unit tests, will be conducted to ensure the application's functionality and user interface consistency.

Documentation and Deployment:

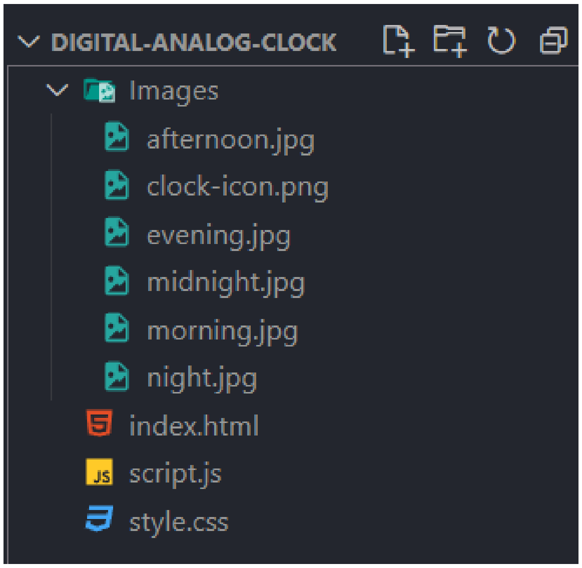
- Detailed documentation, including installation instructions, usage guidelines, and relevant information for developers and users, will be provided.
- The application will be deployed on a web server with a domain, ensuring accessibility and ease of use for end-users.

Dynamic Background:

- The background image of the application will change dynamically based on the current time, creating a visually appealing and immersive experience for the user.
- Different background images will be associated with specific time periods, such as morning, afternoon, evening, and night, to create a seamless and cohesive design.

4.1 File Structure

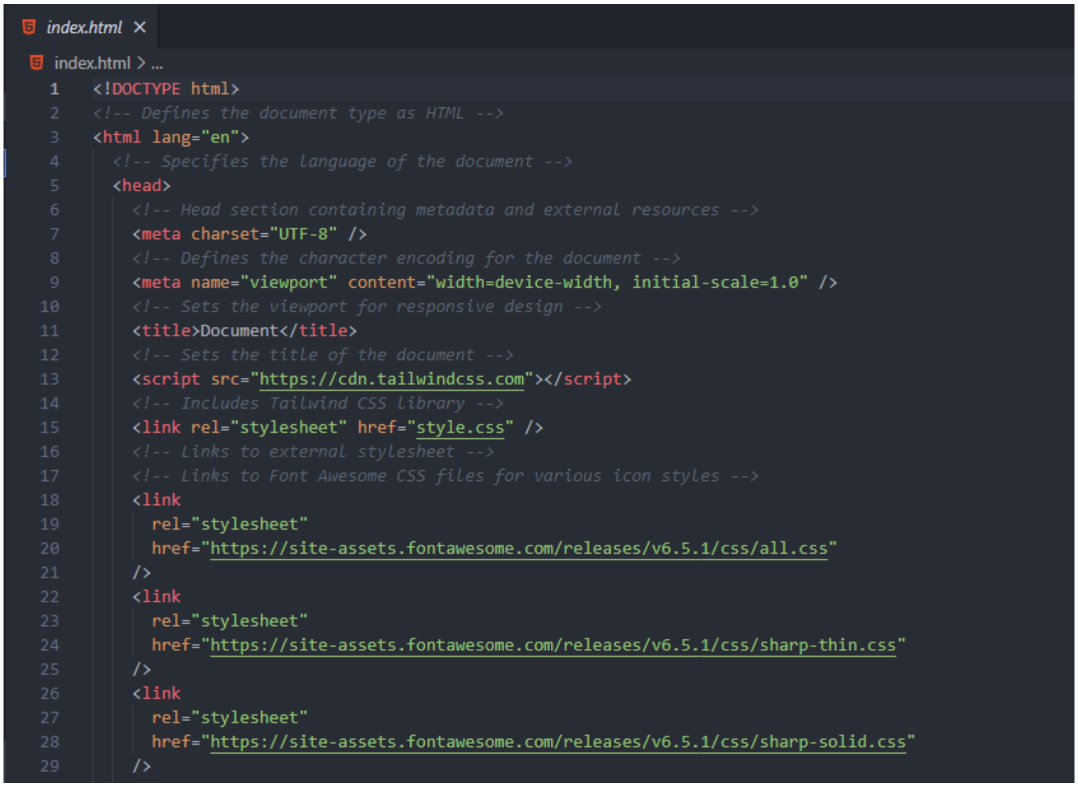
Establishing a meticulous organization of files and folders to uphold uniform file paths and a clutter-free arrangement.



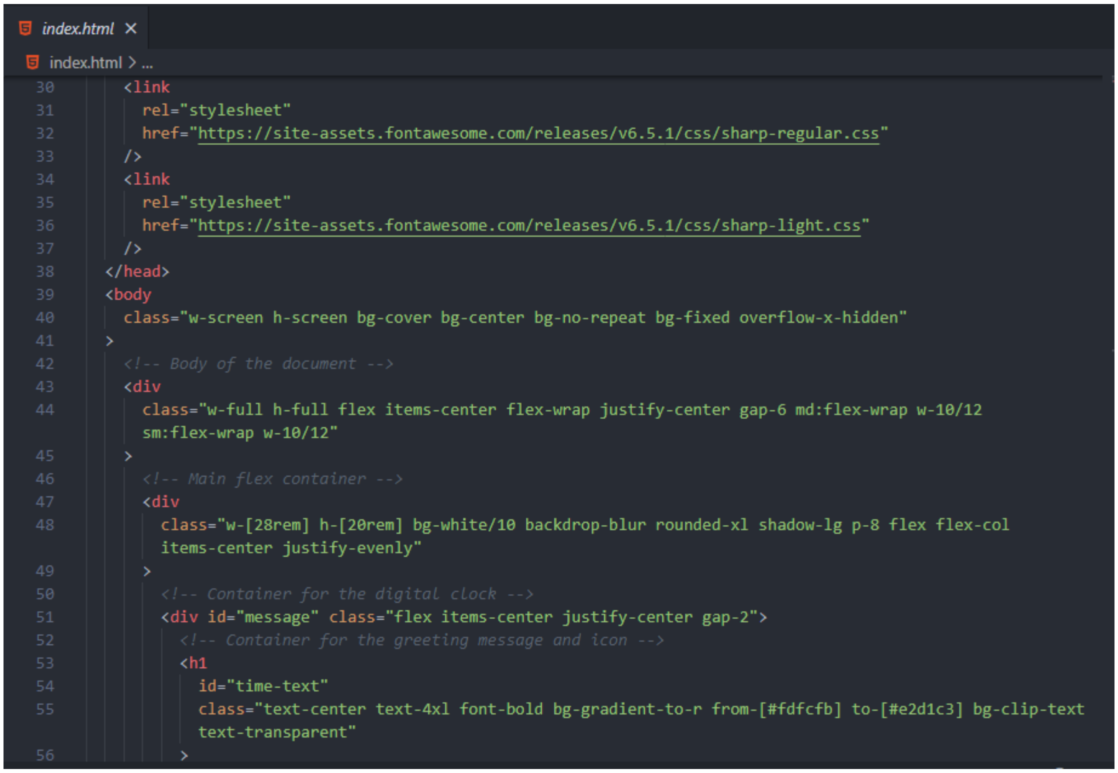
```
DIGITAL-ANALOG-CLOCK
  Images
    afternoon.jpg
    clock-icon.png
    evening.jpg
    midnight.jpg
    morning.jpg
    night.jpg
  index.html
  script.js
  style.css
```

4.2 HTML Code Structure

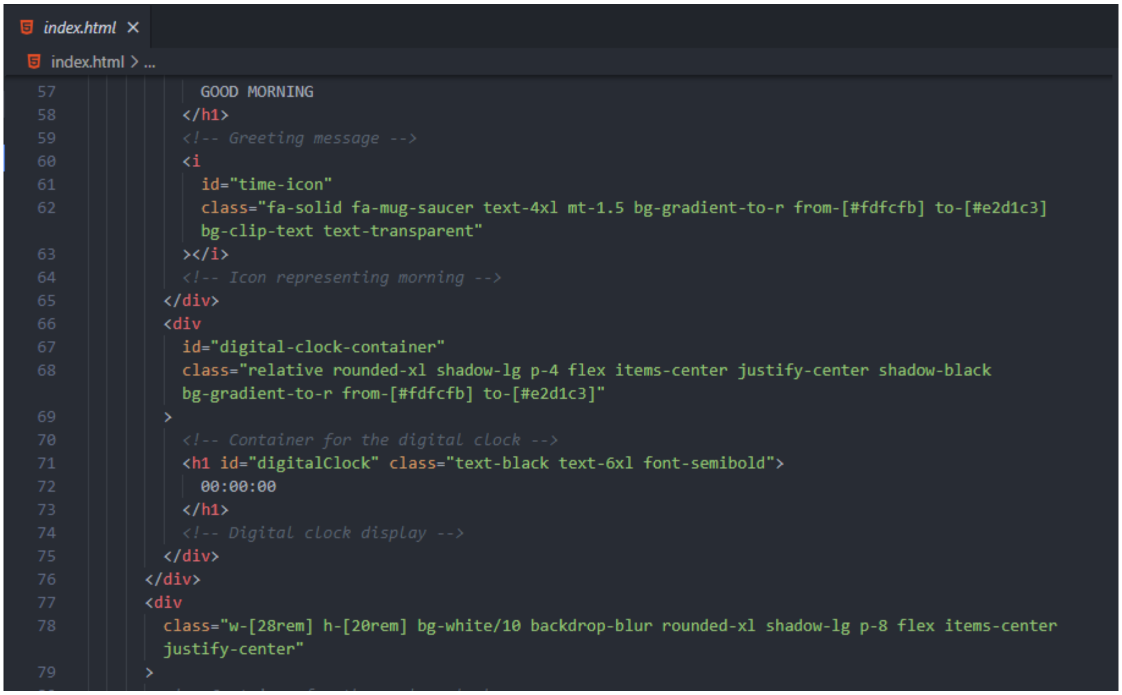
The provided screenshots showcase the HTML code for our Digital and Analog Clock project, depicting the layout and content of our web pages in code format.



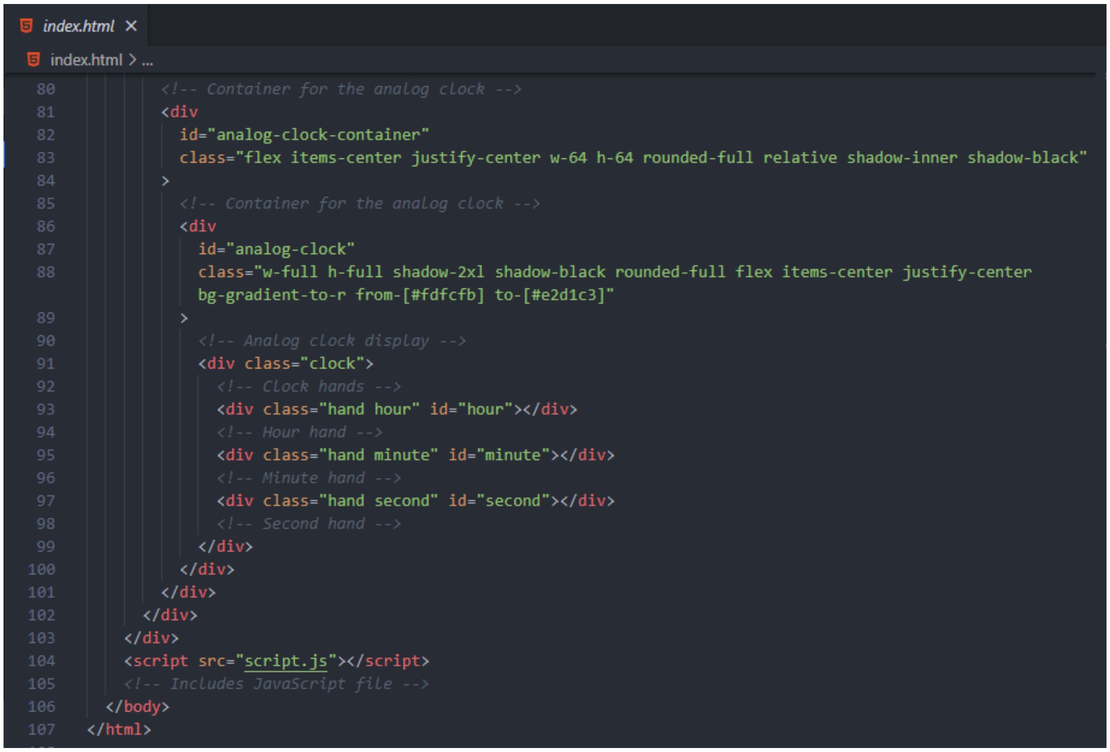
```html
<!DOCTYPE html>
<!-- Defines the document type as HTML -->
<html lang="en">
  <!-- Specifies the language of the document -->
  <head>
    <!-- Head section containing metadata and external resources -->
    <meta charset="UTF-8" />
    <!-- Defines the character encoding for the document -->
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <!-- Sets the viewport for responsive design -->
    <title>Document</title>
    <!-- Sets the title of the document -->
    <script src="https://cdn.tailwindcss.com"></script>
    <!-- Includes Tailwind CSS library -->
    <link rel="stylesheet" href="style.css" />
    <!-- Links to external stylesheet -->
    <!-- Links to Font Awesome CSS files for various icon styles -->
    <link
      rel="stylesheet"
      href="https://site-assets.fontawesome.com/releases/v6.5.1/css/all.css"
    />
    <link
      rel="stylesheet"
      href="https://site-assets.fontawesome.com/releases/v6.5.1/css/sharp-thin.css"
    />
    <link
      rel="stylesheet"
      href="https://site-assets.fontawesome.com/releases/v6.5.1/css/sharp-solid.css"
    />
```



```html
    <link
      rel="stylesheet"
      href="https://site-assets.fontawesome.com/releases/v6.5.1/css/sharp-regular.css"
    />
    <link
      rel="stylesheet"
      href="https://site-assets.fontawesome.com/releases/v6.5.1/css/sharp-light.css"
    />
  </head>
  <body
    class="w-screen h-screen bg-cover bg-center bg-no-repeat bg-fixed overflow-x-hidden"
  >
    <!-- Body of the document -->
    <div
      class="w-full h-full flex items-center flex-wrap justify-center gap-6 md:flex-wrap w-10/12
      sm:flex-wrap w-10/12"
    >
      <!-- Main flex container -->
      <div
        class="w-[28rem] h-[20rem] bg-white/10 backdrop-blur rounded-xl shadow-lg p-8 flex flex-col
        items-center justify-evenly"
      >
        <!-- Container for the digital clock -->
        <div id="message" class="flex items-center justify-center gap-2">
          <!-- Container for the greeting message and icon -->
          <h1
            id="time-text"
            class="text-center text-4xl font-bold bg-gradient-to-r from-[#fdfcfb] to-[#e2d1c3] bg-clip-text
            text-transparent"
          >
```



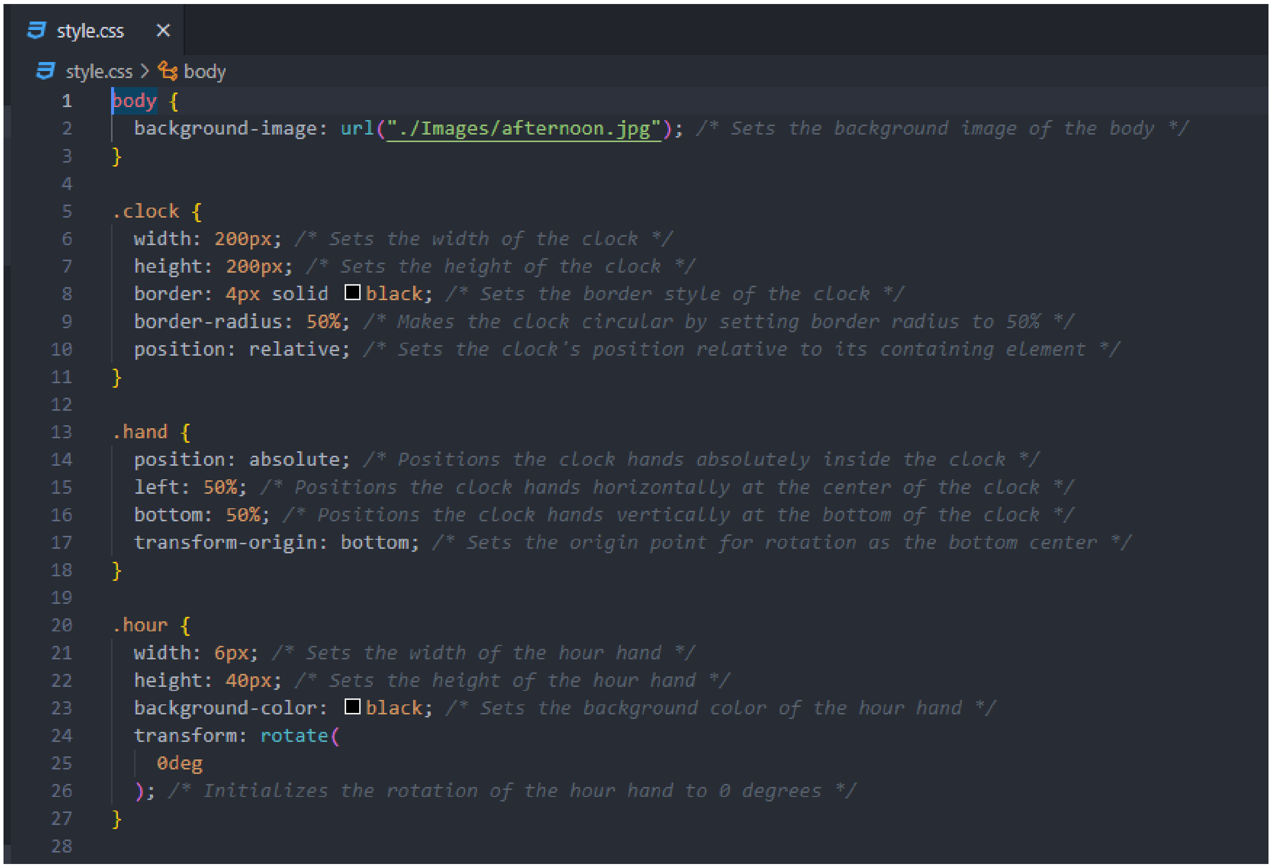
```html
            GOOD MORNING
          </h1>
          <!-- Greeting message -->
          <i
            id="time-icon"
            class="fa-solid fa-mug-saucer text-4xl mt-1.5 bg-gradient-to-r from-[#fdfcfb] to-[#e2d1c3] bg-clip-text text-transparent"
          ></i>
          <!-- Icon representing morning -->
        </div>
        <div
          id="digital-clock-container"
          class="relative rounded-xl shadow-lg p-4 flex items-center justify-center shadow-black bg-gradient-to-r from-[#fdfcfb] to-[#e2d1c3]"
        >
          <!-- Container for the digital clock -->
          <h1 id="digitalClock" class="text-black text-6xl font-semibold">
            00:00:00
          </h1>
          <!-- Digital clock display -->
        </div>
      </div>
      <div
        class="w-[28rem] h-[20rem] bg-white/10 backdrop-blur rounded-xl shadow-lg p-8 flex items-center justify-center"
      >
```



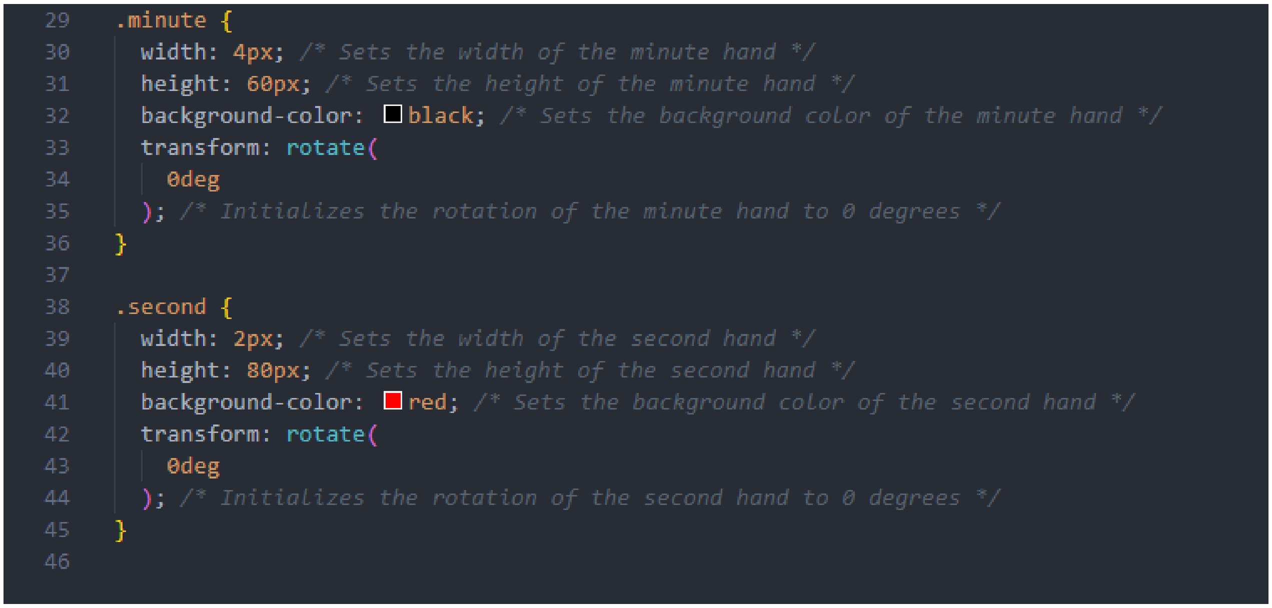
```html
        <!-- Container for the analog clock -->
        <div
          id="analog-clock-container"
          class="flex items-center justify-center w-64 h-64 rounded-full relative shadow-inner shadow-black"
        >
          <!-- Container for the analog clock -->
          <div
            id="analog-clock"
            class="w-full h-full shadow-2xl shadow-black rounded-full flex items-center justify-center bg-gradient-to-r from-[#fdfcfb] to-[#e2d1c3]"
          >
            <!-- Analog clock display -->
            <div class="clock">
              <!-- Clock hands -->
              <div class="hand hour" id="hour"></div>
              <!-- Hour hand -->
              <div class="hand minute" id="minute"></div>
              <!-- Minute hand -->
              <div class="hand second" id="second"></div>
              <!-- Second hand -->
            </div>
          </div>
        </div>
      </div>
    </div>
    <script src="script.js"></script>
    <!-- Includes JavaScript file -->
  </body>
</html>
```

4.3 CSS Code Structure

The screenshots exhibit the CSS code for our Digital and Analog Clock project, demonstrating the styling and visual presentation of our web pages in code format.



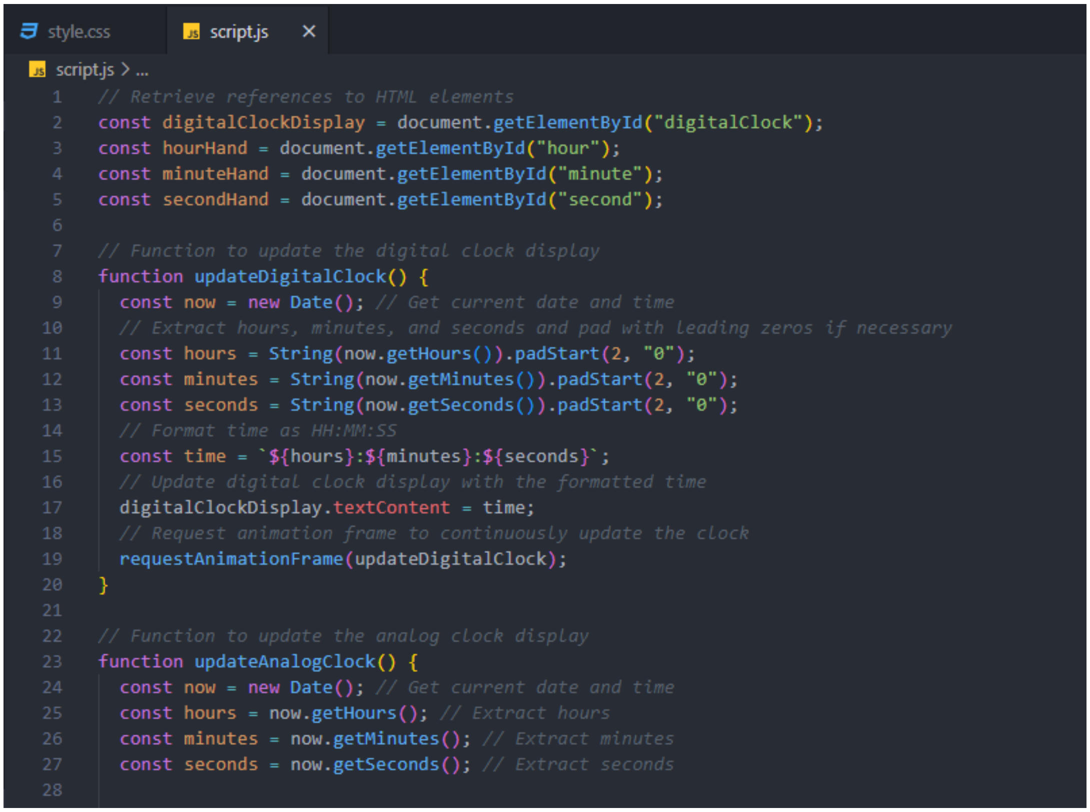
```css
body {
  background-image: url("./Images/afternoon.jpg"); /* Sets the background image of the body */
}

.clock {
  width: 200px; /* Sets the width of the clock */
  height: 200px; /* Sets the height of the clock */
  border: 4px solid black; /* Sets the border style of the clock */
  border-radius: 50%; /* Makes the clock circular by setting border radius to 50% */
  position: relative; /* Sets the clock's position relative to its containing element */
}

.hand {
  position: absolute; /* Positions the clock hands absolutely inside the clock */
  left: 50%; /* Positions the clock hands horizontally at the center of the clock */
  bottom: 50%; /* Positions the clock hands vertically at the bottom of the clock */
  transform-origin: bottom; /* Sets the origin point for rotation as the bottom center */
}

.hour {
  width: 6px; /* Sets the width of the hour hand */
  height: 40px; /* Sets the height of the hour hand */
  background-color: black; /* Sets the background color of the hour hand */
  transform: rotate(
    0deg
  ); /* Initializes the rotation of the hour hand to 0 degrees */
}

```



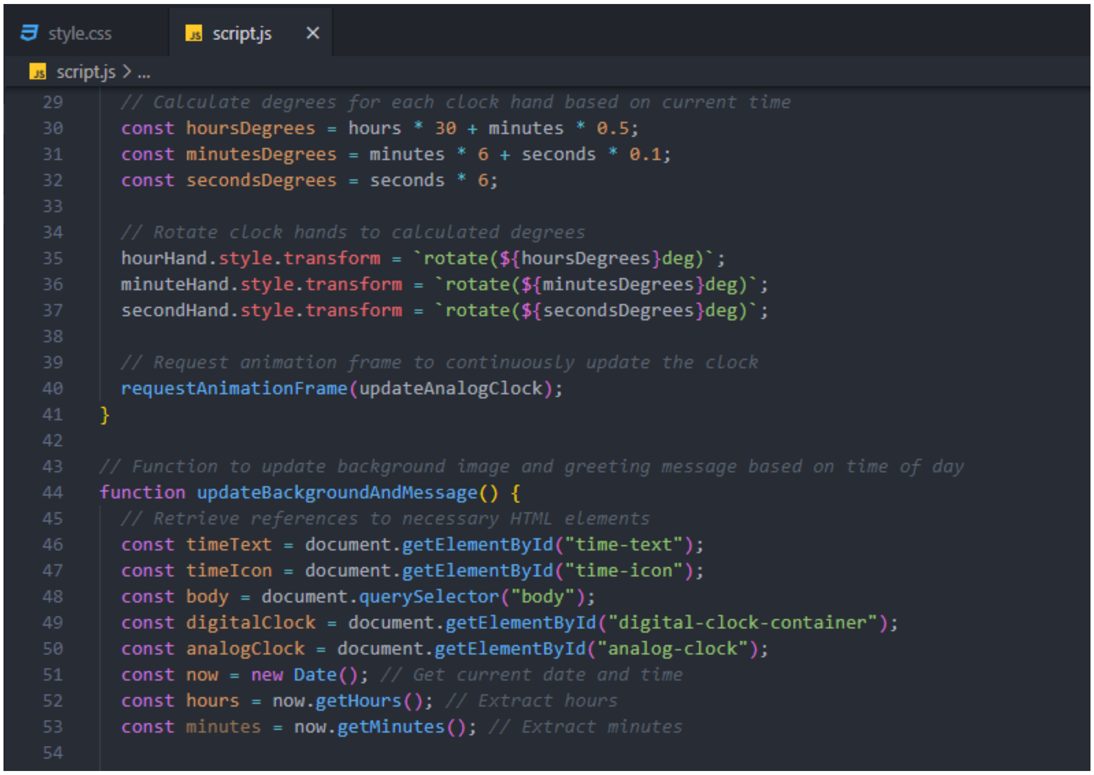
```css
.minute {
  width: 4px; /* Sets the width of the minute hand */
  height: 60px; /* Sets the height of the minute hand */
  background-color: black; /* Sets the background color of the minute hand */
  transform: rotate(
    0deg
  ); /* Initializes the rotation of the minute hand to 0 degrees */
}

.second {
  width: 2px; /* Sets the width of the second hand */
  height: 80px; /* Sets the height of the second hand */
  background-color: red; /* Sets the background color of the second hand */
  transform: rotate(
    0deg
  ); /* Initializes the rotation of the second hand to 0 degrees */
}

```

4.4 Javascript Code Structure

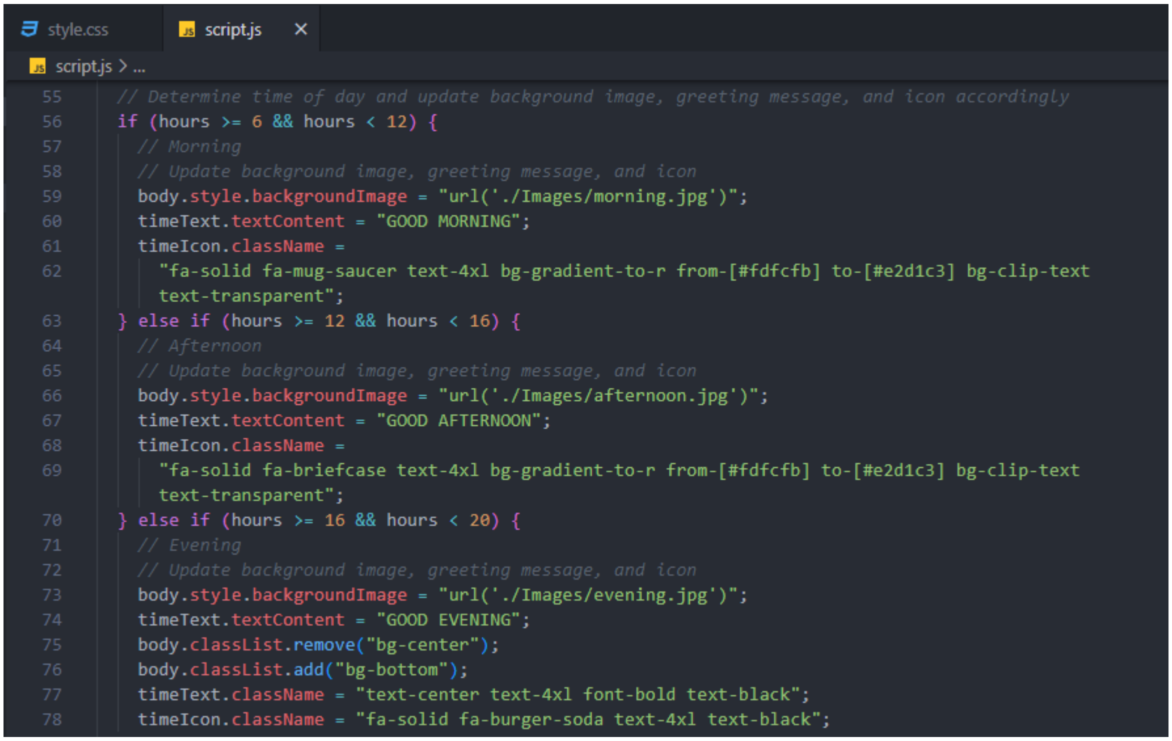
The screenshots depict the JavaScript code for our Digital and Analog Clock project, showcasing the interactive features and dynamic functionalities of our web pages in code format.



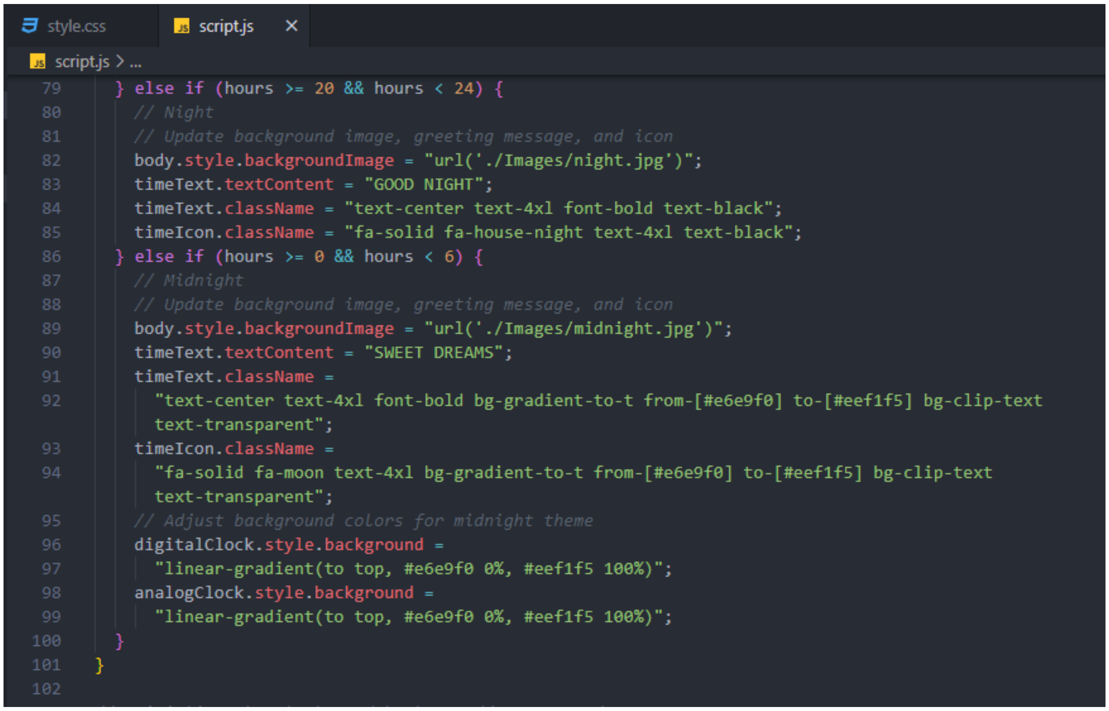
```
// Retrieve references to HTML elements
const digitalClockDisplay = document.getElementById("digitalClock");
const hourHand = document.getElementById("hour");
const minuteHand = document.getElementById("minute");
const secondHand = document.getElementById("second");

// Function to update the digital clock display
function updateDigitalClock() {
  const now = new Date(); // Get current date and time
  // Extract hours, minutes, and seconds and pad with leading zeros if necessary
  const hours = String(now.getHours()).padStart(2, "0");
  const minutes = String(now.getMinutes()).padStart(2, "0");
  const seconds = String(now.getSeconds()).padStart(2, "0");
  // Format time as HH:MM:SS
  const time = `${hours}:${minutes}:${seconds}`;
  // Update digital clock display with the formatted time
  digitalClockDisplay.textContent = time;
  // Request animation frame to continuously update the clock
  requestAnimationFrame(updateDigitalClock);
}

// Function to update the analog clock display
function updateAnalogClock() {
  const now = new Date(); // Get current date and time
  const hours = now.getHours(); // Extract hours
  const minutes = now.getMinutes(); // Extract minutes
  const seconds = now.getSeconds(); // Extract seconds

```



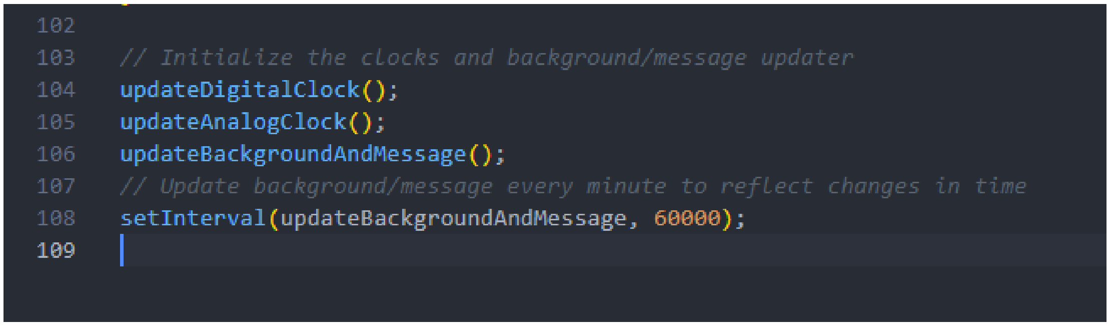
```
  // Calculate degrees for each clock hand based on current time
  const hoursDegrees = hours * 30 + minutes * 0.5;
  const minutesDegrees = minutes * 6 + seconds * 0.1;
  const secondsDegrees = seconds * 6;

  // Rotate clock hands to calculated degrees
  hourHand.style.transform = `rotate(${hoursDegrees}deg)`;
  minuteHand.style.transform = `rotate(${minutesDegrees}deg)`;
  secondHand.style.transform = `rotate(${secondsDegrees}deg)`;

  // Request animation frame to continuously update the clock
  requestAnimationFrame(updateAnalogClock);
}

// Function to update background image and greeting message based on time of day
function updateBackgroundAndMessage() {
  // Retrieve references to necessary HTML elements
  const timeText = document.getElementById("time-text");
  const timeIcon = document.getElementById("time-icon");
  const body = document.querySelector("body");
  const digitalClock = document.getElementById("digital-clock-container");
  const analogClock = document.getElementById("analog-clock");
  const now = new Date(); // Get current date and time
  const hours = now.getHours(); // Extract hours
  const minutes = now.getMinutes(); // Extract minutes

```



```
  // Determine time of day and update background image, greeting message, and icon accordingly
  if (hours >= 6 && hours < 12) {
    // Morning
    // Update background image, greeting message, and icon
    body.style.backgroundImage = "url('./Images/morning.jpg')";
    timeText.textContent = "GOOD MORNING";
    timeIcon.className =
      "fa-solid fa-mug-saucer text-4xl bg-gradient-to-r from-[#fdfcfb] to-[#e2d1c3] bg-clip-text
      text-transparent";
  } else if (hours >= 12 && hours < 16) {
    // Afternoon
    // Update background image, greeting message, and icon
    body.style.backgroundImage = "url('./Images/afternoon.jpg')";
    timeText.textContent = "GOOD AFTERNOON";
    timeIcon.className =
      "fa-solid fa-briefcase text-4xl bg-gradient-to-r from-[#fdfcfb] to-[#e2d1c3] bg-clip-text
      text-transparent";
  } else if (hours >= 16 && hours < 20) {
    // Evening
    // Update background image, greeting message, and icon
    body.style.backgroundImage = "url('./Images/evening.jpg')";
    timeText.textContent = "GOOD EVENING";
    body.classList.remove("bg-center");
    body.classList.add("bg-bottom");
    timeText.className = "text-center text-4xl font-bold text-black";
    timeIcon.className = "fa-solid fa-burger-soda text-4xl text-black";
```



```javascript
  } else if (hours >= 20 && hours < 24) {
    // Night
    // Update background image, greeting message, and icon
    body.style.backgroundImage = "url('./Images/night.jpg')";
    timeText.textContent = "GOOD NIGHT";
    timeText.className = "text-center text-4xl font-bold text-black";
    timeIcon.className = "fa-solid fa-house-night text-4xl text-black";
  } else if (hours >= 0 && hours < 6) {
    // Midnight
    // Update background image, greeting message, and icon
    body.style.backgroundImage = "url('./Images/midnight.jpg')";
    timeText.textContent = "SWEET DREAMS";
    timeText.className =
      "text-center text-4xl font-bold bg-gradient-to-t from-[#e6e9f0] to-[#eef1f5] bg-clip-text text-transparent";
    timeIcon.className =
      "fa-solid fa-moon text-4xl bg-gradient-to-t from-[#e6e9f0] to-[#eef1f5] bg-clip-text text-transparent";
    // Adjust background colors for midnight theme
    digitalClock.style.background =
      "linear-gradient(to top, #e6e9f0 0%, #eef1f5 100%)";
    analogClock.style.background =
      "linear-gradient(to top, #e6e9f0 0%, #eef1f5 100%)";
  }
}

```



```javascript

// Initialize the clocks and background/message updater
updateDigitalClock();
updateAnalogClock();
updateBackgroundAndMessage();
// Update background/message every minute to reflect changes in time
setInterval(updateBackgroundAndMessage, 60000);

```

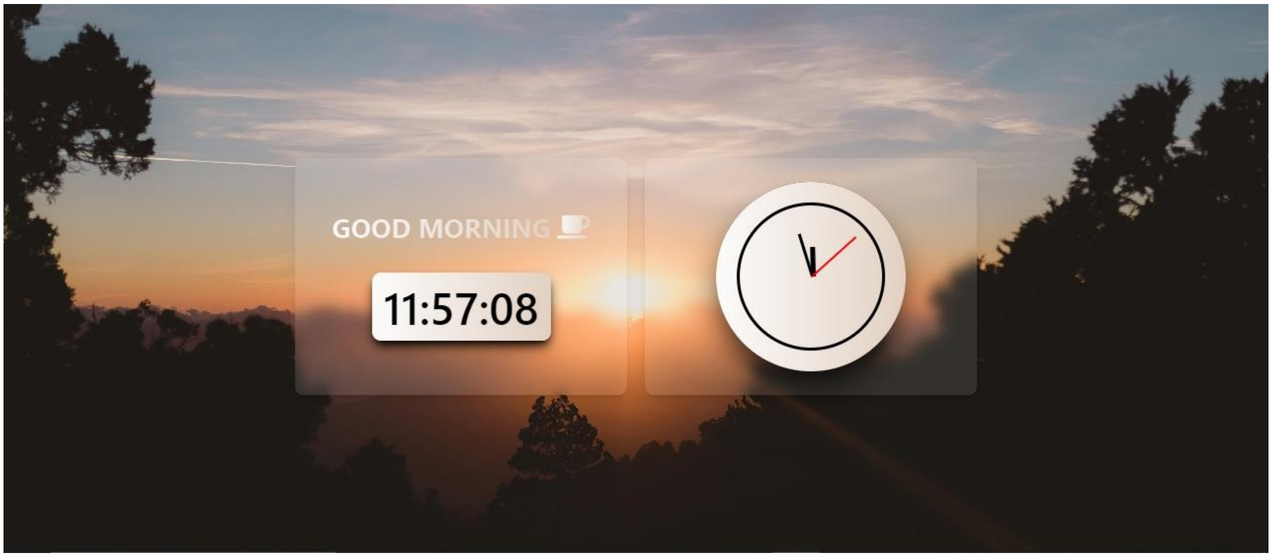
5. Results

The displayed results showcase the synchronized digital and analog clocks, alongside dynamically changing background images and greeting messages based on the current time. Through rigorous testing and user feedback, the following key outcomes have been achieved:

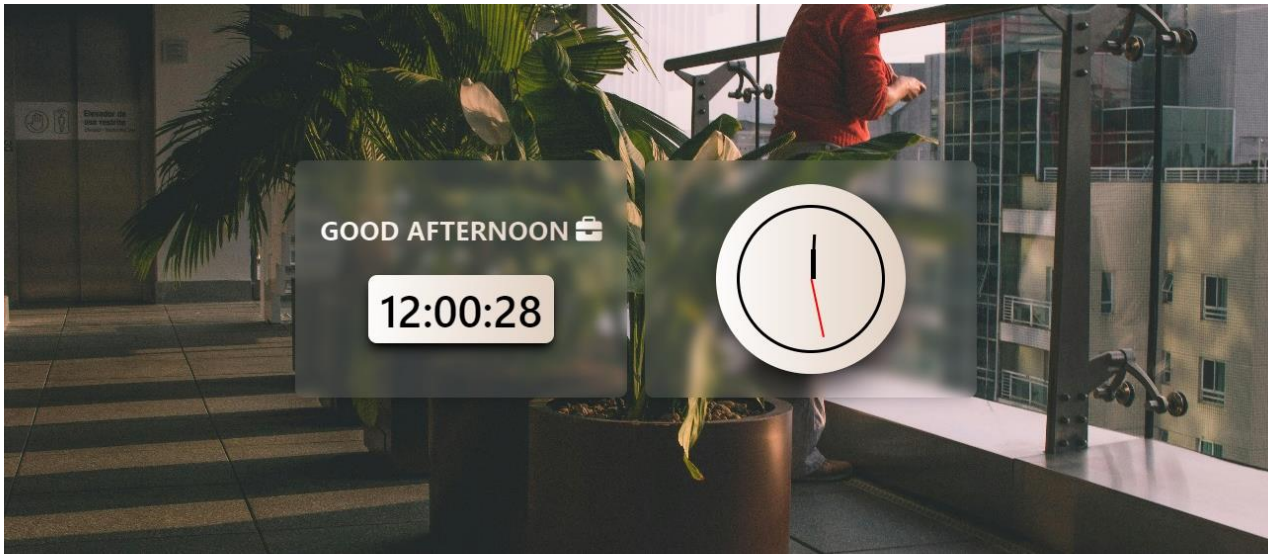
Git Hub Repository Link: https://github.com/itsrudra143/Digital-and-Analog-Clock

Git Hub Pages Link: https://itsrudra143.github.io/Digital-and-Analog-Clock/

These screenshots capture different instances of my project, showcasing its appearance and functionality at various points in time.



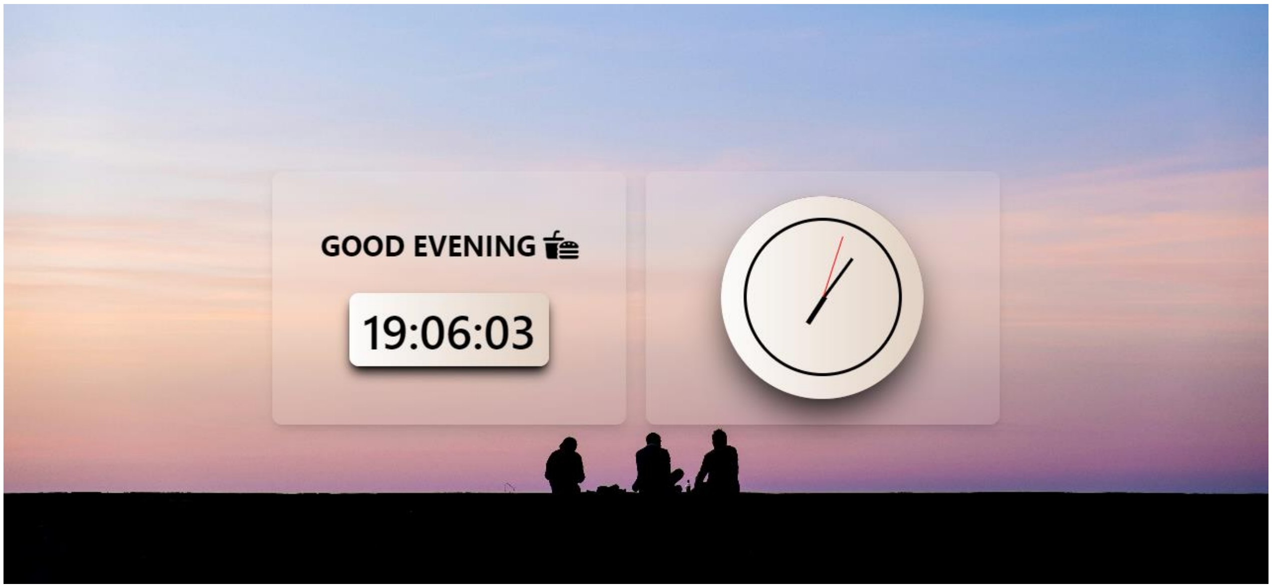
Morning Ambiance



GOOD AFTERNOON 💼

12:00:28

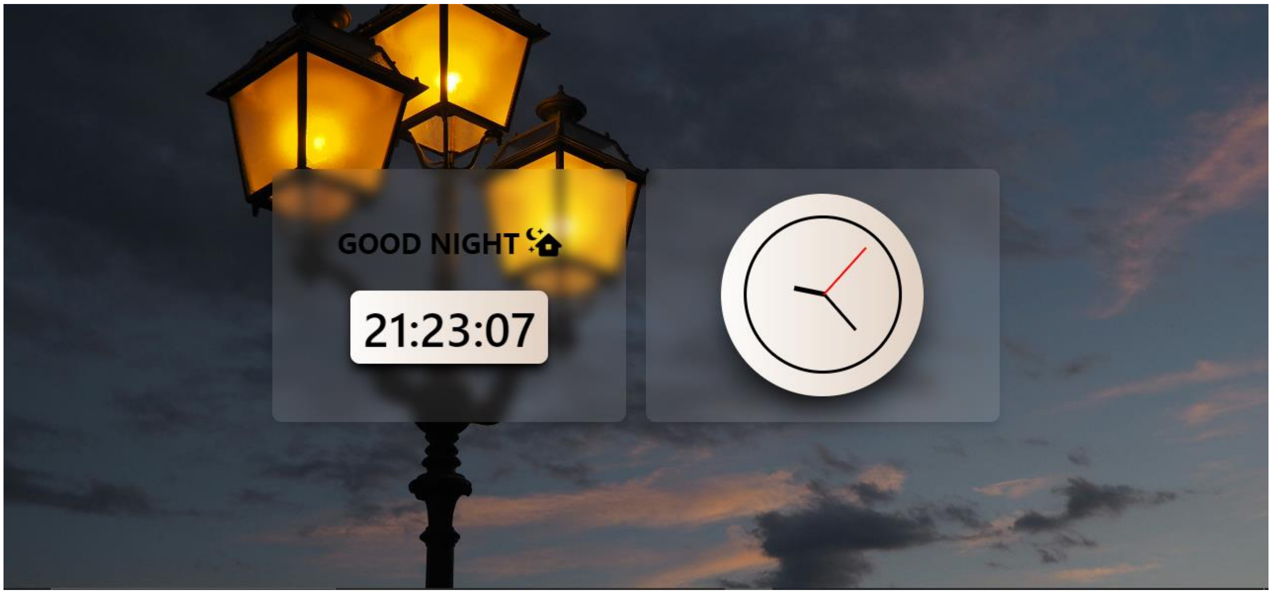
Afternoon Serenity



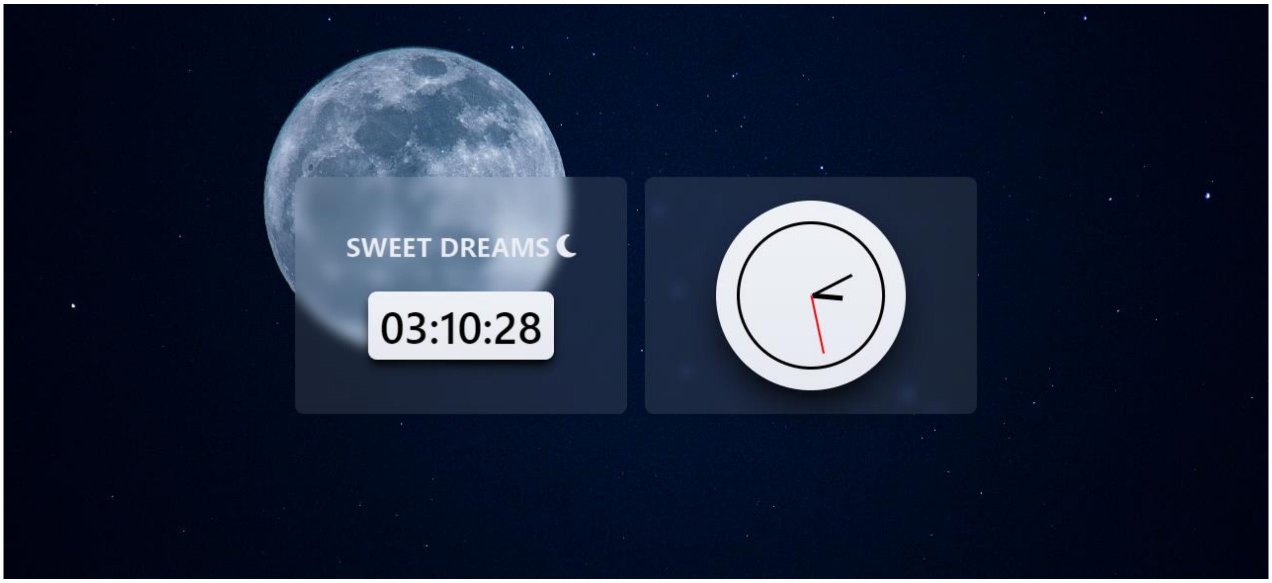
GOOD EVENING 🍔

19:06:03

Evening Tranquility



Nighttime Stillness



Midnight Mystery

- Our project focuses on digital and analog clocks, presenting accurate time
- Utilizing HTML, CSS, Tailwind, and JavaScript, we've crafted a seamless and visually appealing time-telling interface.
- JavaScript ensures synchronization between the digital and analog clocks, offering precise timekeeping.
- A standout feature of our project is its adaptability—the background image changes dynamically to reflect the time of day.
- Whether it's the serene glow of morning, the warm hues of afternoon, the calming twilight of evening, the tranquil darkness of night, or the enigmatic ambiance of midnight, our project adjusts the background image accordingly.
- Through this integration of technology and design, our project offers both functionality and aesthetic appeal, enhancing the user experience with each passing hour.

6. References

- **Mozilla Developer Network (MDN) - HTML, CSS, JS Documentation:**

  Website: https://developer.mozilla.org/

  Description: MDN offers comprehensive documentation on HTML, CSS, and JavaScript, covering everything from basic syntax to advanced concepts and APIs.

- **W3Schools - HTML, CSS, JavaScript Tutorials:**

  Website: https://www.w3schools.com/

  Description: W3Schools provides beginner-friendly tutorials and references for HTML, CSS, and JavaScript, along with interactive code examples.

- **Tailwind CSS Documentation:**

  Website: https://tailwindcss.com/docs

  Description: The official Tailwind CSS documentation provides detailed guidance on using Tailwind CSS for building modern and responsive web interfaces. It includes utility classes, customization options, and best practices.

- **CSS-Tricks:**

  Website: https://css-tricks.com/

  Description: CSS-Tricks is a web design community that offers articles, tutorials, and resources related to CSS, HTML, and frontend development. It covers a wide range of topics, including layout techniques, CSS animations, and frontend frameworks.

- **Smashing Magazine:**

  Website**:** https://www.smashingmagazine.com/

  Description: Smashing Magazine publishes articles, tutorials, and case studies on web design and development, including frontend technologies like HTML, CSS, JavaScript, and modern frameworks like Tailwind CSS. It also features interviews with industry experts and showcases of cutting-edge web projects.