VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)
CHENNAI

WEB PROGRAMMING LAB

BCSE203E

SLOT: L11 + L12 + L19 + L20

LAB MANUAL

Exercise 6

Date: 02-03-2025

GITHUB REPO LINK:

https://github.com/divyanshupatel17/23BAI1214_WEB_LAB

WEB PROG LAB PORTFOLIO:

https://23bai1214-divyanshu-web-v1.netlify.app/

https://23bai1214-divyanshu-web-v2.netlify.app/

https://23bai1214-divyanshu-web-v3.netlify.app/

SUBMITTED TO:

PROF. DR. ASHOKA RAJAN R

SUBMITTED BY:

DIVYANSHU PATEL
23BAI1214

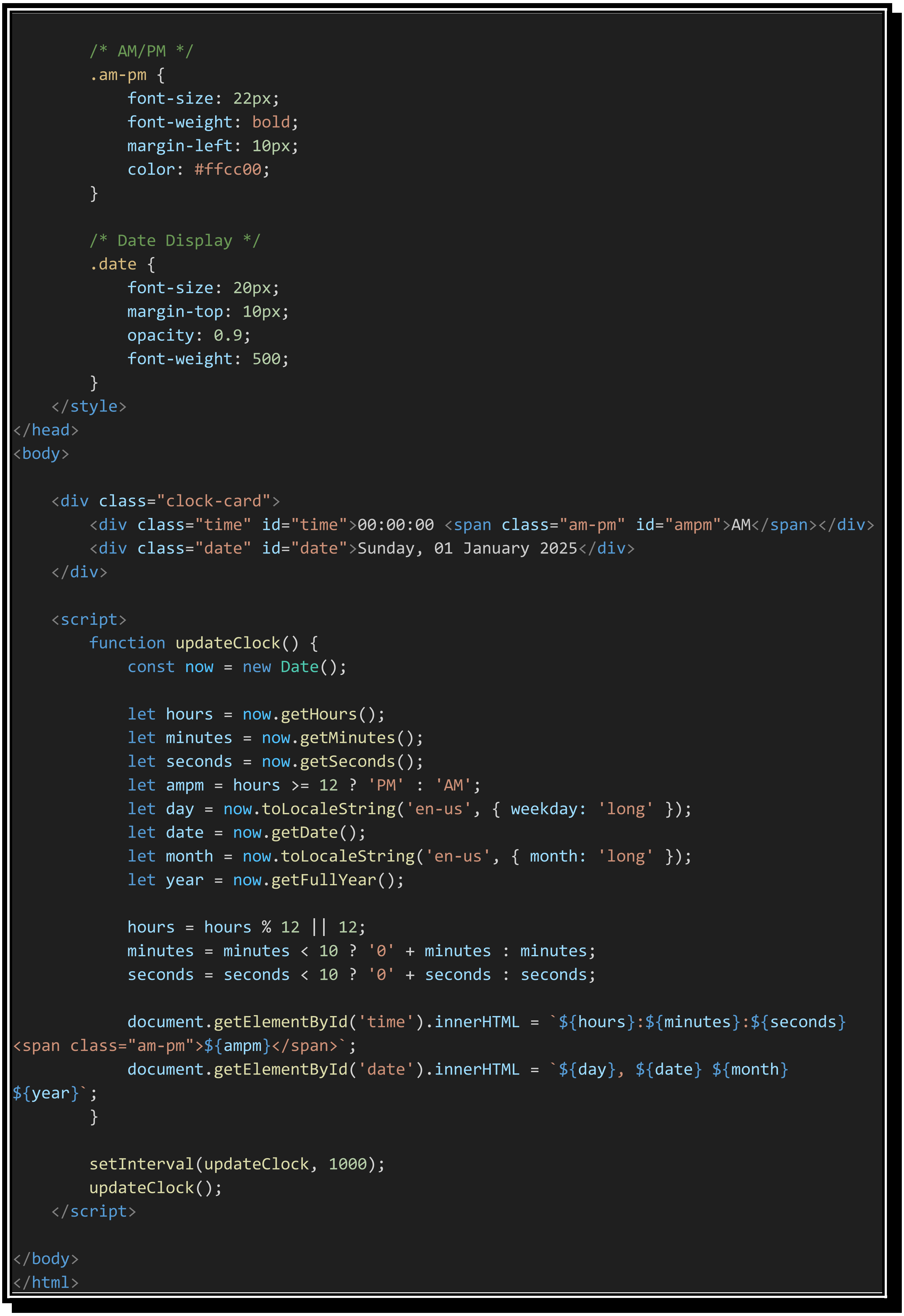
School of Computer Science and Engineering

Vellore Institute of Technology, Chennai

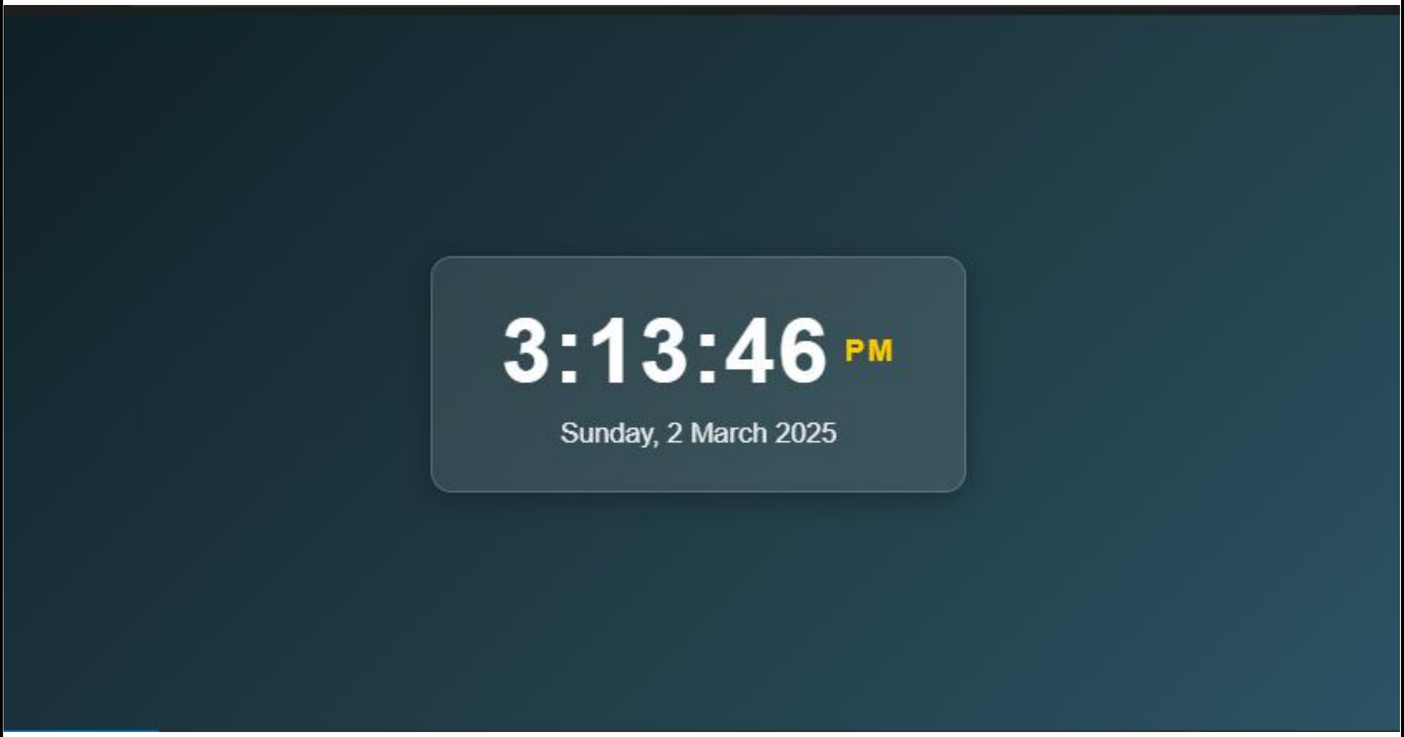
1. Digital Clock

https://23bai1214-divyanshu-exercise-6-t-1.netlify.app/

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Digital Clock</title>
    <style>
        * {
            margin: 0;
            padding: 0;
            box-sizing: border-box;
            font-family: 'Poppins', sans-serif;
        }
        body {
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
            background: linear-gradient(135deg, #0f2027, #203a43, #2c5364);
            color: #fff;
            text-align: center;
        }

        /* Clock Card */
        .clock-card {
            background: rgba(255, 255, 255, 0.1);
            backdrop-filter: blur(10px);
            padding: 30px 50px;
            border-radius: 15px;
            box-shadow: 0px 5px 20px rgba(0, 0, 0, 0.3);
            border: 1px solid rgba(255, 255, 255, 0.2);
            text-align: center;
            min-width: 300px;
        }

        /* Time Display */
        .time {
            font-size: 65px;
            font-weight: bold;
            letter-spacing: 2px;
            display: flex;
            justify-content: center;
            align-items: center;
        }
```



```
        /* AM/PM */
        .am-pm {
            font-size: 22px;
            font-weight: bold;
            margin-left: 10px;
            color: #ffcc00;
        }

        /* Date Display */
        .date {
            font-size: 20px;
            margin-top: 10px;
            opacity: 0.9;
            font-weight: 500;
        }
    </style>
</head>
<body>

    <div class="clock-card">
        <div class="time" id="time">00:00:00 <span class="am-pm" id="ampm">AM</span></div>
        <div class="date" id="date">Sunday, 01 January 2025</div>
    </div>

    <script>
        function updateClock() {
            const now = new Date();

            let hours = now.getHours();
            let minutes = now.getMinutes();
            let seconds = now.getSeconds();
            let ampm = hours >= 12 ? 'PM' : 'AM';
            let day = now.toLocaleString('en-us', { weekday: 'long' });
            let date = now.getDate();
            let month = now.toLocaleString('en-us', { month: 'long' });
            let year = now.getFullYear();

            hours = hours % 12 || 12;
            minutes = minutes < 10 ? '0' + minutes : minutes;
            seconds = seconds < 10 ? '0' + seconds : seconds;

            document.getElementById('time').innerHTML = `${hours}:${minutes}:${seconds}
<span class="am-pm">${ampm}</span>`;
            document.getElementById('date').innerHTML = `${day}, ${date} ${month}
${year}`;
        }

        setInterval(updateClock, 1000);
        updateClock();
    </script>

</body>
</html>
```

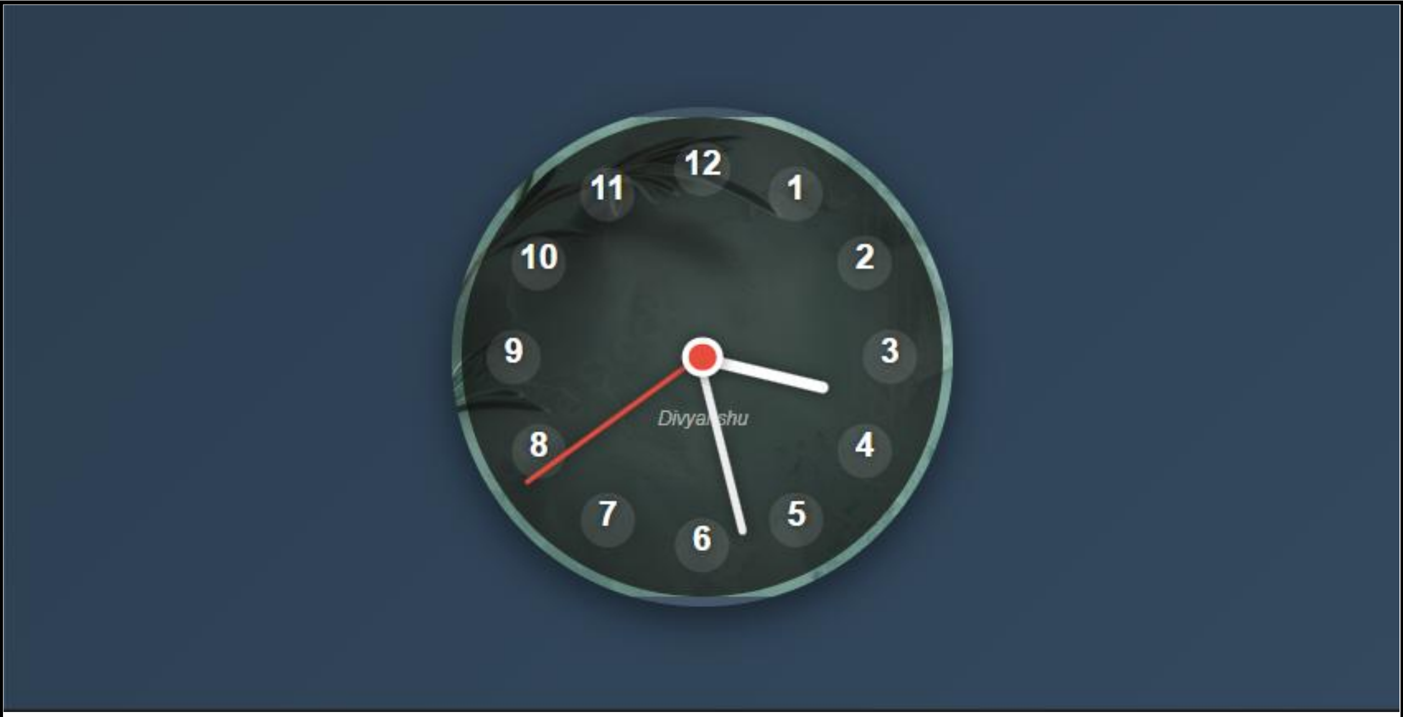


2. Analog Clock

https://23bai1214-divyanshu-exercise-6-t-2.netlify.app/

```
<!-- 2. Analog Clock -->

<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>Analog Clock</title>

        <style>
            body {
                display: flex;
                justify-content: center;
                align-items: center;
                min-height: 100vh;
                margin: 0;
                background: linear-gradient(135deg, #2c3e50, #34495e);
                font-family: 'Arial', sans-serif;
            }
            .clock {
                width: 350px;
                height: 350px;
                border: 8px solid rgba(255, 255, 255, 0.1);
                border-radius: 50%;
                background: url('bg.png') no-repeat center center/cover;
                position: relative;
                box-shadow: 0 10px 30px rgba(0, 0, 0, 0.5);
            }
            .clock::before {
                content: '';
                position: absolute;
                top: 0;
                left: 0;
                width: 100%;
                height: 100%;
                background: radial-gradient(circle, rgba(40, 40, 40, 0.7) 30%, rgba(20,
20, 20, 0.8) 90%);
                border-radius: 50%;
            }
            .number {
                position: absolute;
                width: 100%;
                height: 100%;
                text-align: center;
                font-size: 1.6rem;
                font-weight: bold;
                color: white;
                padding: 18px;
                box-sizing: border-box;
            }
            .number div {
                position: relative;
                display: inline-block;
                background: rgba(255, 255, 255, 0.1);
                padding: 4px;
                border-radius: 50%;
                width: 32px;
                height: 32px;
                line-height: 24px;
                text-align: center;
                text-shadow: 0 0 5px rgba(0, 0, 0, 0.5);
            }
            .hand {
                position: absolute;
                bottom: 50%;
                left: 50%;
                transform-origin: bottom;
                border-radius: 8px;
                z-index: 10;
                box-shadow: 0 0 10px rgba(0, 0, 0, 0.5);
            }
            .hour {
                width: 8px;
                height: 28%;
                background: #fff;
                margin-left: -4px;
            }
            .minute {
                width: 6px;
                height: 38%;
                background: #eee;
                margin-left: -3px;
            }
            .second {
                width: 3px;
                height: 45%;
                background: #e74c3c;
                margin-left: -1.5px;
            }
            .center {
                position: absolute;
                width: 20px;
                height: 20px;
                background: #e74c3c;
                border: 5px solid #fff;
                border-radius: 50%;
                top: 50%;
                left: 50%;
                transform: translate(-50%, -50%);
                z-index: 20;
                box-shadow: 0 0 10px rgba(0, 0, 0, 0.5);
            }
            .brand {
                position: absolute;
                bottom: 35%;
                width: 100%;
                text-align: center;
                color: rgba(255, 255, 255, 0.6);
                font-size: 14px;
                font-style: italic;
                z-index: 5;
            }
        </style>
    </head>

    <body>
        <div class="clock">
            <div class="hand hour"></div>
            <div class="hand minute"></div>
            <div class="hand second"></div>
            <div class="center"></div>
            <div class="brand">Divyanshu</div>
        </div>

        <script>
            function setupClock() {
                const numbers = [12, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11];
                const clock = document.querySelector('.clock');

                numbers.forEach((number, index) => {
                    const element = document.createElement('div');
                    element.className = 'number';
                    element.style.transform = `rotate(${index * 30}deg)`;
```

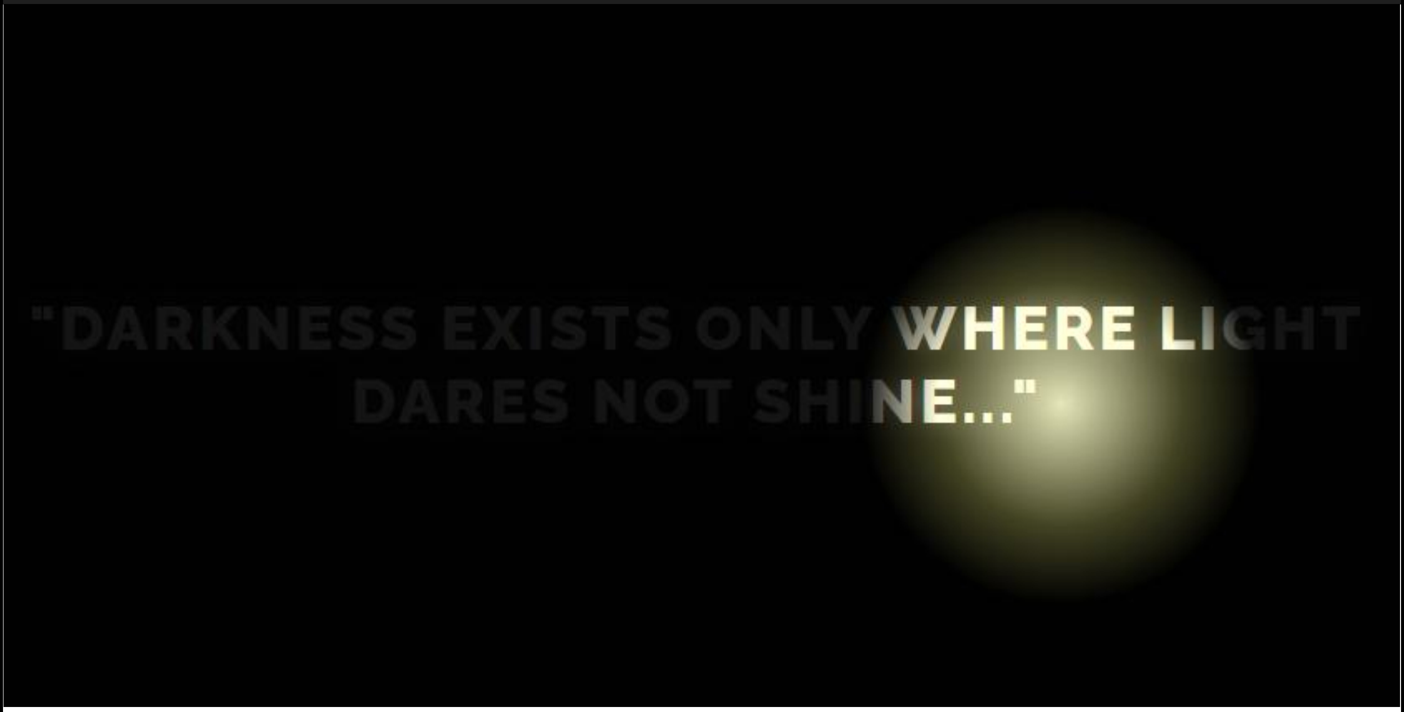
```
                    const text = document.createElement('div');
                    text.style.transform = `rotate(-${index * 30}deg)`;
                    text.textContent = number;

                    element.appendChild(text);
                    clock.appendChild(element);
                });
            }

            function updateClock() {
                const now = new Date();
                const seconds = now.getSeconds();
                const minutes = now.getMinutes();
                const hours = now.getHours() % 12;

                const secondDeg = (seconds / 60) * 360;
                const minuteDeg = ((minutes + seconds / 60) / 60) * 360;
                const hourDeg = ((hours + minutes / 60) / 12) * 360;

                document.querySelector('.second').style.transform = `translateX(-50%) ro-
tate(${secondDeg}deg)`;
                document.querySelector('.minute').style.transform = `translateX(-50%) ro-
tate(${minuteDeg}deg)`;
                document.querySelector('.hour').style.transform = `translateX(-50%) ro-
tate(${hourDeg}deg)`;
            }

            setupClock();
            setInterval(updateClock, 1000);
            updateClock();
        </script>
    </body>
</html>
```



3. Flashlight Text Reveal

https://23bai1214-divyanshu-exercise-6-t-3.netlify.app/

```
<!-- 3. Flashlight Text Reveal -->

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Flashlight Text Reveal</title>

    <style>
        @import url('https://fonts.googleapis.com/css2?family=Raleway:wght@800&dis-
play=swap');

        body {
            margin: 0;
            height: 100vh;
            display: flex;
            align-items: center;
            justify-content: center;
            background-color: #111;
            overflow: hidden;
            color: white;
            font-family: 'Raleway', sans-serif;
            text-align: center;
            cursor: none;
            position: relative;
        }
        .container {
            position: relative;
            font-size: 2.8rem;
            font-weight: bold;
            text-transform: uppercase;
            letter-spacing: 3px;
            text-shadow: 0 0 15px rgba(255, 255, 255, 0.3);
        }

        .flashlight {
            position: absolute;
            top: 0;
            left: 0;
            width: 100%;
            height: 100%;
            background: radial-gradient(circle 180px at var(--x, 50%) var(--y, 50%),
                        rgba(255, 255, 204, 0.9),
                        rgba(255, 255, 102, 0.2) 50%,
                        rgba(0, 0, 0, 0.92) 80%);
            pointer-events: none;
            transition: background 0.1s ease-out;
        }

        .torch {
            position: absolute;
            width: 60px;
            height: 60px;
            background: url('https://upload.wikimedia.org/wikipedia/commons/3/3b/Flash-
light_icon.png') no-repeat center;
            background-size: contain;
            pointer-events: none;
            transform: translate(-50%, -50%);
            opacity: 0.8;
            filter: drop-shadow(0 0 10px rgba(255, 255, 255, 0.5));
        }
    </style>
</head>

<body>
    <div class="container">"Darkness exists only where light dares not shine..."</div>
    <div class="flashlight"></div>
    <div class="torch"></div>

    <script>
        const flashlight = document.querySelector('.flashlight');
        const torch = document.querySelector('.torch');

        document.addEventListener('mousemove', (e) => {
            flashlight.style.setProperty('--x', `${e.clientX}px`);
            flashlight.style.setProperty('--y', `${e.clientY}px`);

            torch.style.left = `${e.clientX}px`;
            torch.style.top = `${e.clientY}px`;
        });
    </script>
</body>
</html>
```

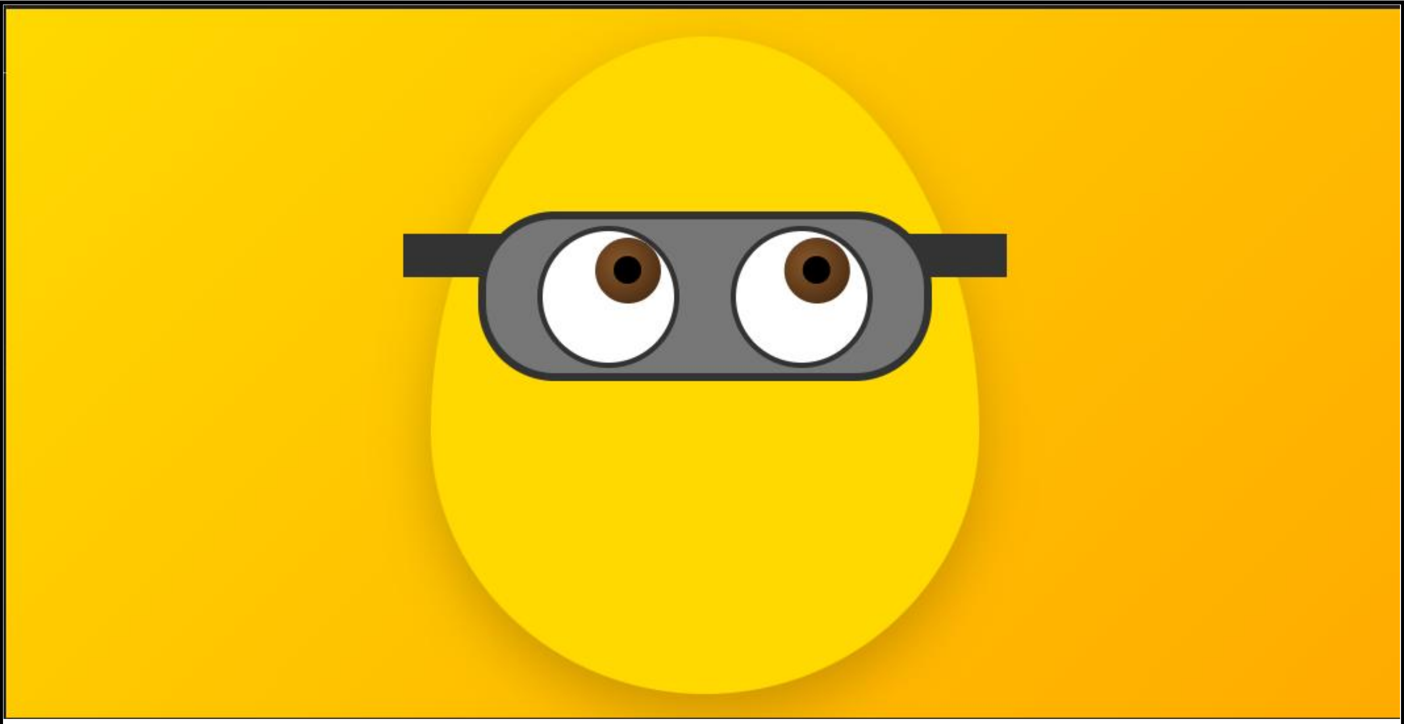


4. Minion Eye

https://23bai1214-divyanshu-exercise-6-t-4.netlify.app/

```
<!-- 4. Minion Eye -->

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Minion Eye Tracker</title>
    <style>
        body {
            margin: 0;
            height: 100vh;
            display: flex;
            justify-content: center;
            align-items: center;
            background: linear-gradient(135deg, #ffd900, #ffab00);
            overflow: hidden;
        }
        .minion-face {
            position: relative;
            width: 500px;
            height: 600px;
            background: #ffd900;
            border-radius: 50% / 60% 60% 40% 40%;
            box-shadow: 0 20px 50px rgba(0, 0, 0, 0.2);
            display: flex;
            justify-content: center;
            align-items: center;
        }
        .goggles-strap {
            position: absolute;
            top: 180px;
            width: 550px;
            height: 40px;
            background: #333;
        }
        .goggles-container {
            position: absolute;
            top: 160px;
            width: 400px;
            height: 140px;
            background: #777;
            border-radius: 70px;
            border: 8px solid #333;
            display: flex;
            justify-content: space-evenly;
            align-items: center;
        }
        .eye-container {
            position: relative;
            width: 120px;
            height: 120px;
            background: #fff;
            border-radius: 50%;
            border: 5px solid #333;
            display: flex;
            justify-content: center;
            align-items: center;
            overflow: hidden;
        }
        .eye {
            position: relative;
            width: 60px;
            height: 60px;
            background: radial-gradient(circle at 40% 40%, #8b5a2b 0%, #4b2c14 80%);
            border-radius: 50%;
        }
        .pupil {
            position: absolute;
            width: 25px;
            height: 25px;
            background: #000;
            border-radius: 50%;
            top: 17px;
            left: 17px;
        }
        .custom-cursor {
            position: fixed;
```

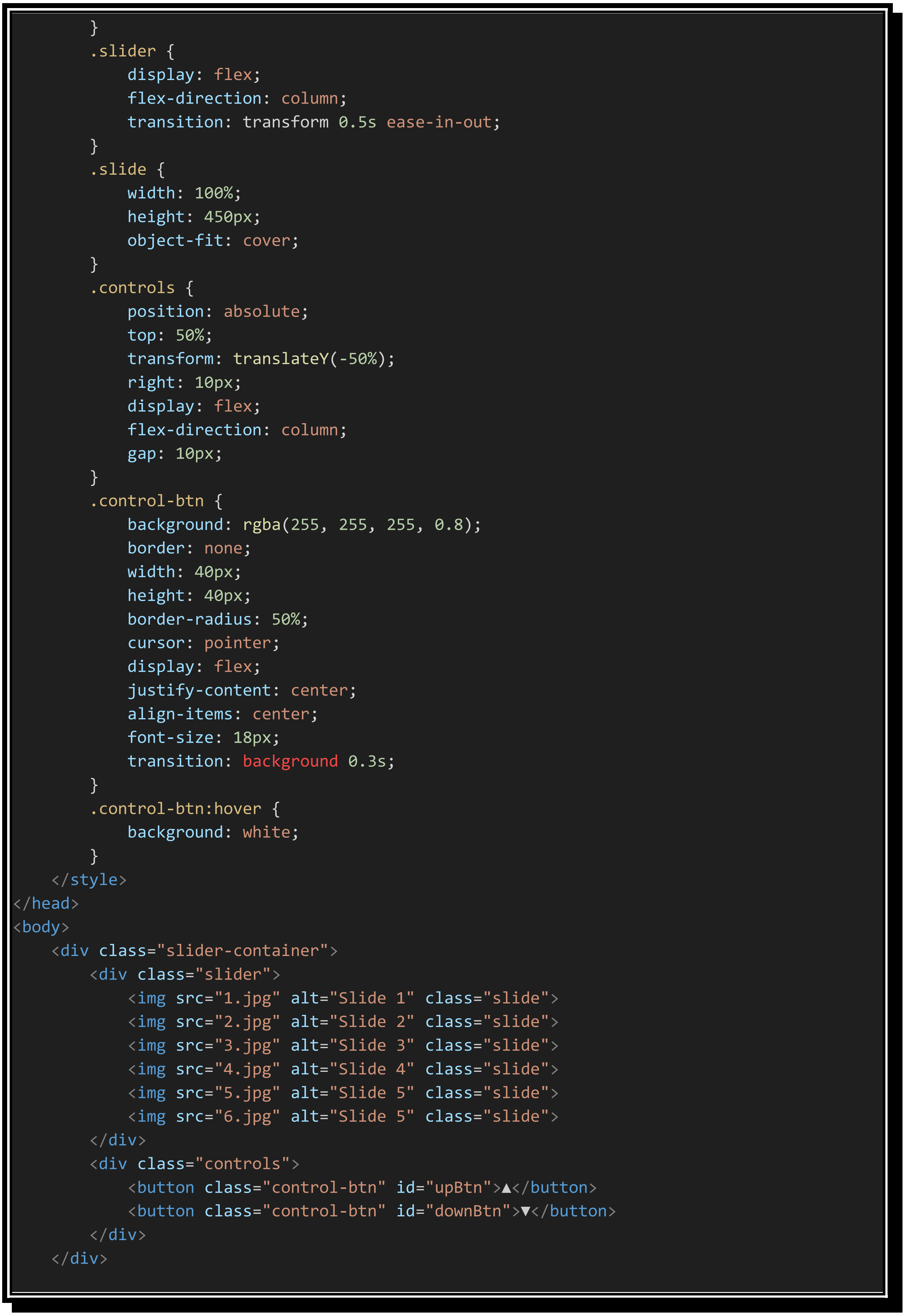
```
            width: 24px;
            height: 24px;
            border-radius: 50%;
            background: rgba(255, 255, 255, 0.3);
            border: 3px solid #333;
            transform: translate(-50%, -50%);
            z-index: 9999;
        }
    </style>
</head>
<body>
    <div class="custom-cursor" id="customCursor"></div>
    <div class="minion-face">
        <div class="goggles-strap"></div>
        <div class="goggles-container">
            <div class="eye-container">
                <div class="eye" id="leftEye">
                    <div class="pupil"></div>
                </div>
            </div>
            <div class="eye-container">
                <div class="eye" id="rightEye">
                    <div class="pupil"></div>
                </div>
            </div>
        </div>
    </div>
    <script>
        document.addEventListener('mousemove', function(e) {
            document.getElementById('customCursor').style.transform = `translate(${e.cli-
entX}px, ${e.clientY}px)`;
            moveEyes(e.clientX, e.clientY);
        });
        function moveEyes(x, y) {
            document.querySelectorAll('.eye').forEach(eye => {
                const rect = eye.getBoundingClientRect(), dx = x - (rect.left + rect.width
/ 2), dy = y - (rect.top + rect.height / 2), maxMove = 18;
                eye.style.transform = `translate(${Math.min(dx / 6, maxMove)}px,
${Math.min(dy / 6, maxMove)}px)`;
            });
        }
    </script>
</body>
</html>
```



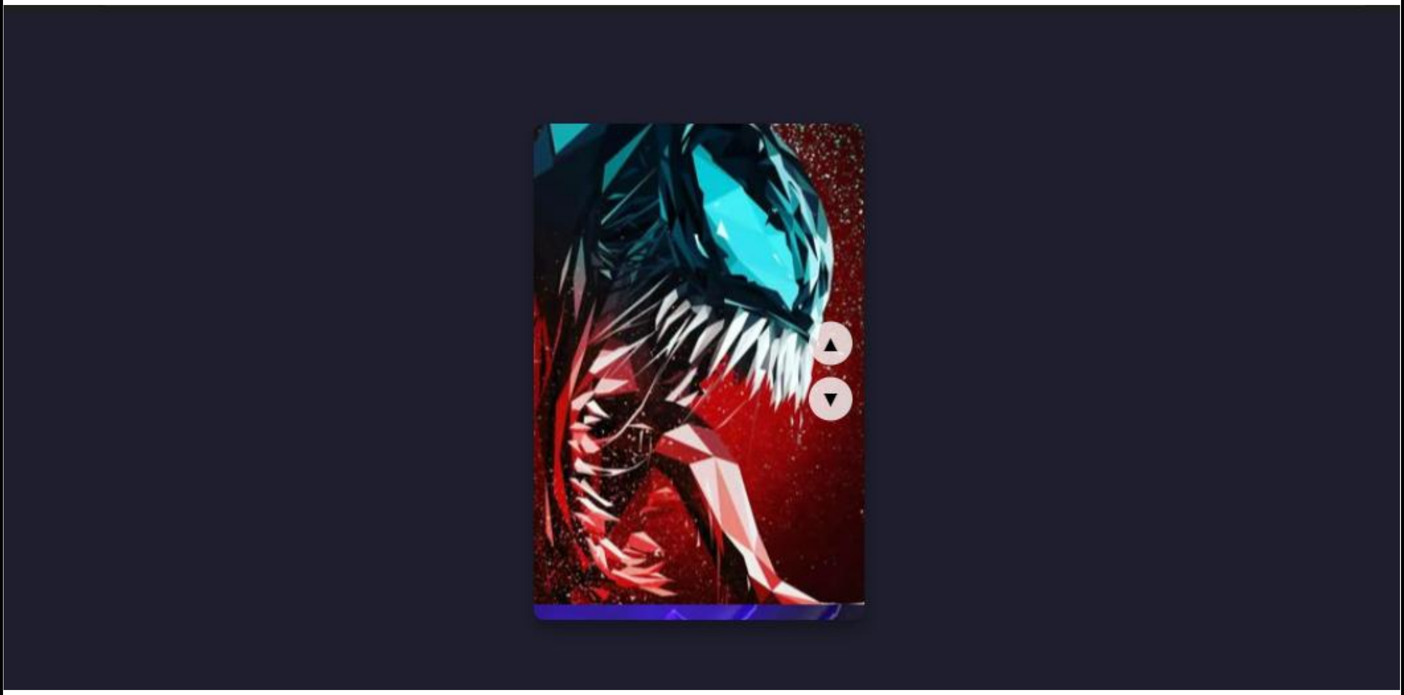
5. Vertical Image Slider

https://23bai1214-divyanshu-exercise-6-t-5.netlify.app/

```
<!-- 5. Vertical Image Slider  -->

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Vertical Image Slider</title>
    <style>
        * {
            margin: 0;
            padding: 0;
            box-sizing: border-box;
        }
        body {
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
            background-color: #1e1e2e;
            font-family: Arial, sans-serif;
        }
        .slider-container {
            position: relative;
            width: 300px;
            height: 450px;
            overflow: hidden;
            border-radius: 10px;
            box-shadow: 0 8px 16px rgba(0, 0, 0, 0.3);
```



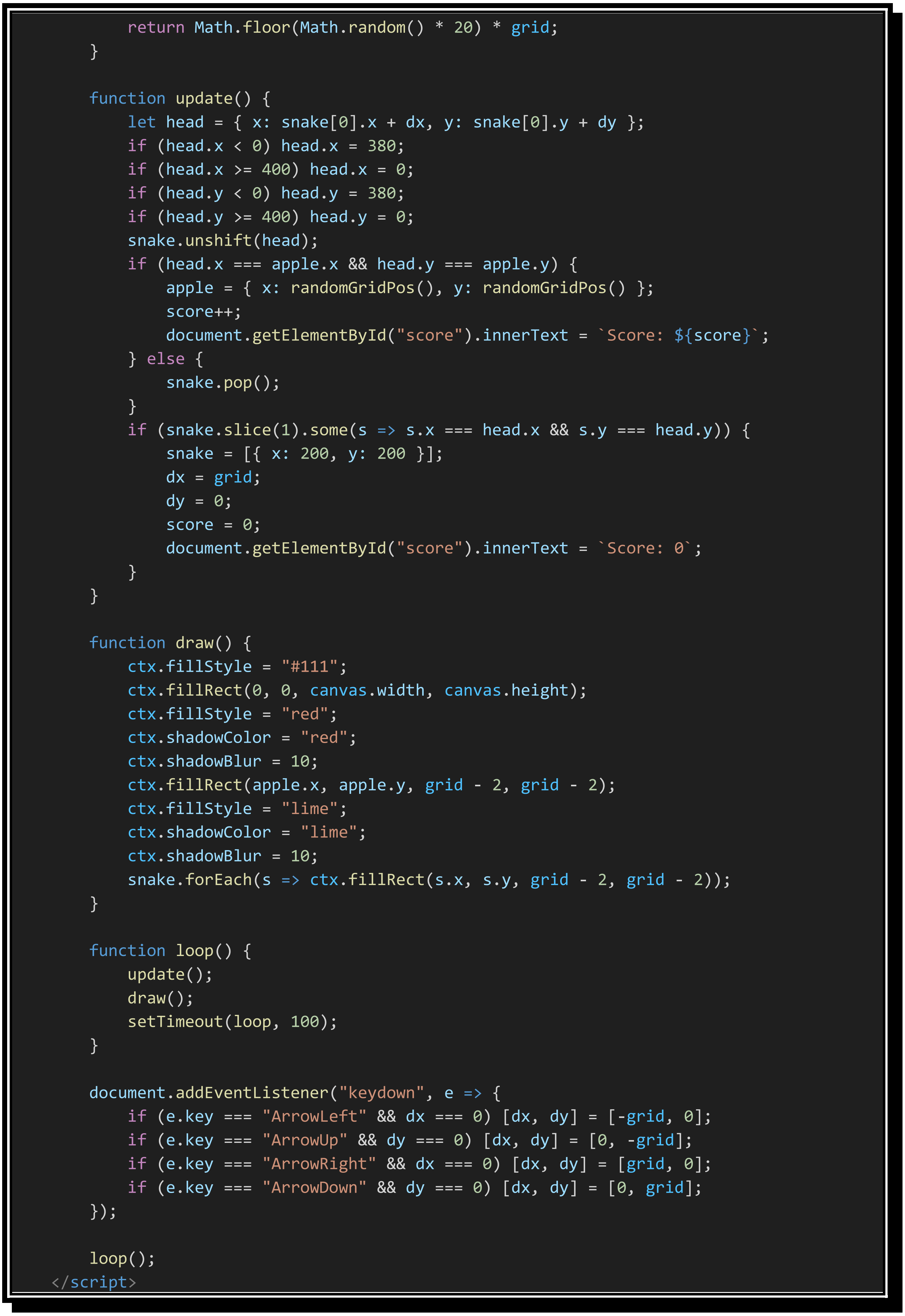
```
        }
        .slider {
            display: flex;
            flex-direction: column;
            transition: transform 0.5s ease-in-out;
        }
        .slide {
            width: 100%;
            height: 450px;
            object-fit: cover;
        }
        .controls {
            position: absolute;
            top: 50%;
            transform: translateY(-50%);
            right: 10px;
            display: flex;
            flex-direction: column;
            gap: 10px;
        }
        .control-btn {
            background: rgba(255, 255, 255, 0.8);
            border: none;
            width: 40px;
            height: 40px;
            border-radius: 50%;
            cursor: pointer;
            display: flex;
            justify-content: center;
            align-items: center;
            font-size: 18px;
            transition: background 0.3s;
        }
        .control-btn:hover {
            background: white;
        }
    </style>
</head>
<body>
    <div class="slider-container">
        <div class="slider">
            <img src="1.jpg" alt="Slide 1" class="slide">
            <img src="2.jpg" alt="Slide 2" class="slide">
            <img src="3.jpg" alt="Slide 3" class="slide">
            <img src="4.jpg" alt="Slide 4" class="slide">
            <img src="5.jpg" alt="Slide 5" class="slide">
            <img src="6.jpg" alt="Slide 5" class="slide">
        </div>
        <div class="controls">
            <button class="control-btn" id="upBtn">▲</button>
            <button class="control-btn" id="downBtn">▼</button>
        </div>
    </div>
    <script>
        const slider = document.querySelector('.slider');
        const slides = document.querySelectorAll('.slide');
        const upBtn = document.getElementById('upBtn');
        const downBtn = document.getElementById('downBtn');
        let index = 0;
        const slideHeight = slides[0].clientHeight;

        function updateSlider() {
            slider.style.transform = `translateY(${-index * slideHeight}px)`;
        }

        upBtn.addEventListener('click', () => {
            index = index > 0 ? index - 1 : slides.length - 1;
            updateSlider();
        });

        downBtn.addEventListener('click', () => {
            index = index < slides.length - 1 ? index + 1 : 0;
            updateSlider();
        });

        setInterval(() => {
            index = index < slides.length - 1 ? index + 1 : 0;
            updateSlider();
        }, 5000);
    </script>
</body>
</html>
```



6. Snake Game

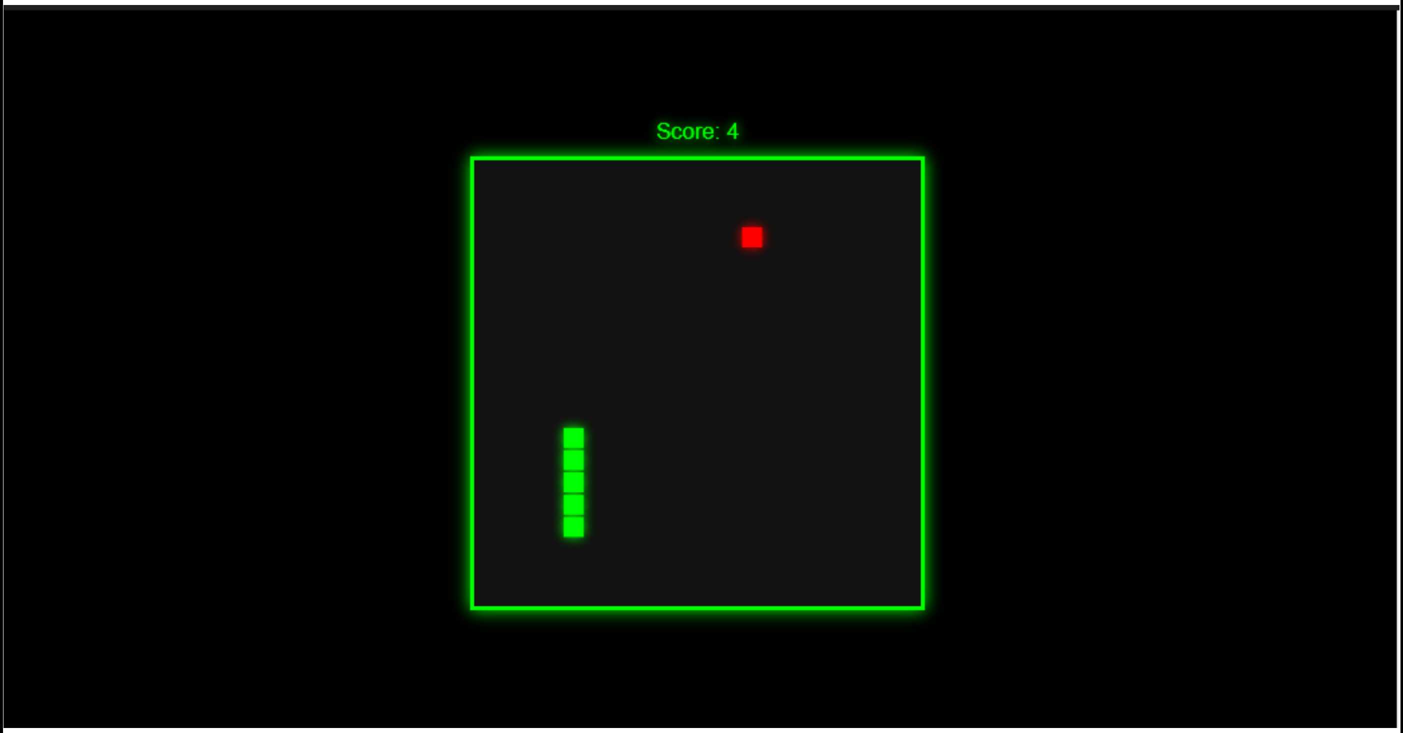
https://23bai1214-divyanshu-exercise-6-t-6.netlify.app/

```html
<!-- 6. Snake Game  -->

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Snake Game</title>
    <style>
        @font-face {
            font-family: 'Pixel';
            src:
url('https://fonts.gstatic.com/s/pressstart2p/v8/e3t4euUuV4RGabz3VmsTuyzpbpORTq6aaj-
07Q.ttf');
        }
        body {
            display: flex;
            flex-direction: column;
            align-items: center;
            justify-content: center;
            height: 100vh;
            background: black;
            color: lime;
            font-family: 'Pixel', sans-serif;
            text-align: center;
        }
        canvas {
            border: 4px solid lime;
            box-shadow: 0 0 15px lime;
            background: #111;
        }
        #score {
            font-size: 20px;
            margin-bottom: 10px;
            text-shadow: 0 0 10px lime;
        }
    </style>
</head>
<body>
    <div id="score">Score: 0</div>
    <canvas id="game" width="400" height="400"></canvas>

    <script>
        const canvas = document.getElementById("game");
        const ctx = canvas.getContext("2d");
        const grid = 20;
        let snake = [{ x: 200, y: 200 }];
        let apple = { x: 300, y: 300 };
        let dx = grid, dy = 0, score = 0;

        function randomGridPos() {
```



```
            return Math.floor(Math.random() * 20) * grid;
        }

        function update() {
            let head = { x: snake[0].x + dx, y: snake[0].y + dy };
            if (head.x < 0) head.x = 380;
            if (head.x >= 400) head.x = 0;
            if (head.y < 0) head.y = 380;
            if (head.y >= 400) head.y = 0;
            snake.unshift(head);
            if (head.x === apple.x && head.y === apple.y) {
                apple = { x: randomGridPos(), y: randomGridPos() };
                score++;
                document.getElementById("score").innerText = `Score: ${score}`;
            } else {
                snake.pop();
            }
            if (snake.slice(1).some(s => s.x === head.x && s.y === head.y)) {
                snake = [{ x: 200, y: 200 }];
                dx = grid;
                dy = 0;
                score = 0;
                document.getElementById("score").innerText = `Score: 0`;
            }
        }

        function draw() {
            ctx.fillStyle = "#111";
            ctx.fillRect(0, 0, canvas.width, canvas.height);
            ctx.fillStyle = "red";
            ctx.shadowColor = "red";
            ctx.shadowBlur = 10;
            ctx.fillRect(apple.x, apple.y, grid - 2, grid - 2);
            ctx.fillStyle = "lime";
            ctx.shadowColor = "lime";
            ctx.shadowBlur = 10;
            snake.forEach(s => ctx.fillRect(s.x, s.y, grid - 2, grid - 2));
        }

        function loop() {
            update();
            draw();
            setTimeout(loop, 100);
        }

        document.addEventListener("keydown", e => {
            if (e.key === "ArrowLeft" && dx === 0) [dx, dy] = [-grid, 0];
            if (e.key === "ArrowUp" && dy === 0) [dx, dy] = [0, -grid];
            if (e.key === "ArrowRight" && dx === 0) [dx, dy] = [grid, 0];
            if (e.key === "ArrowDown" && dy === 0) [dx, dy] = [0, grid];
        });

        loop();
    </script>
```

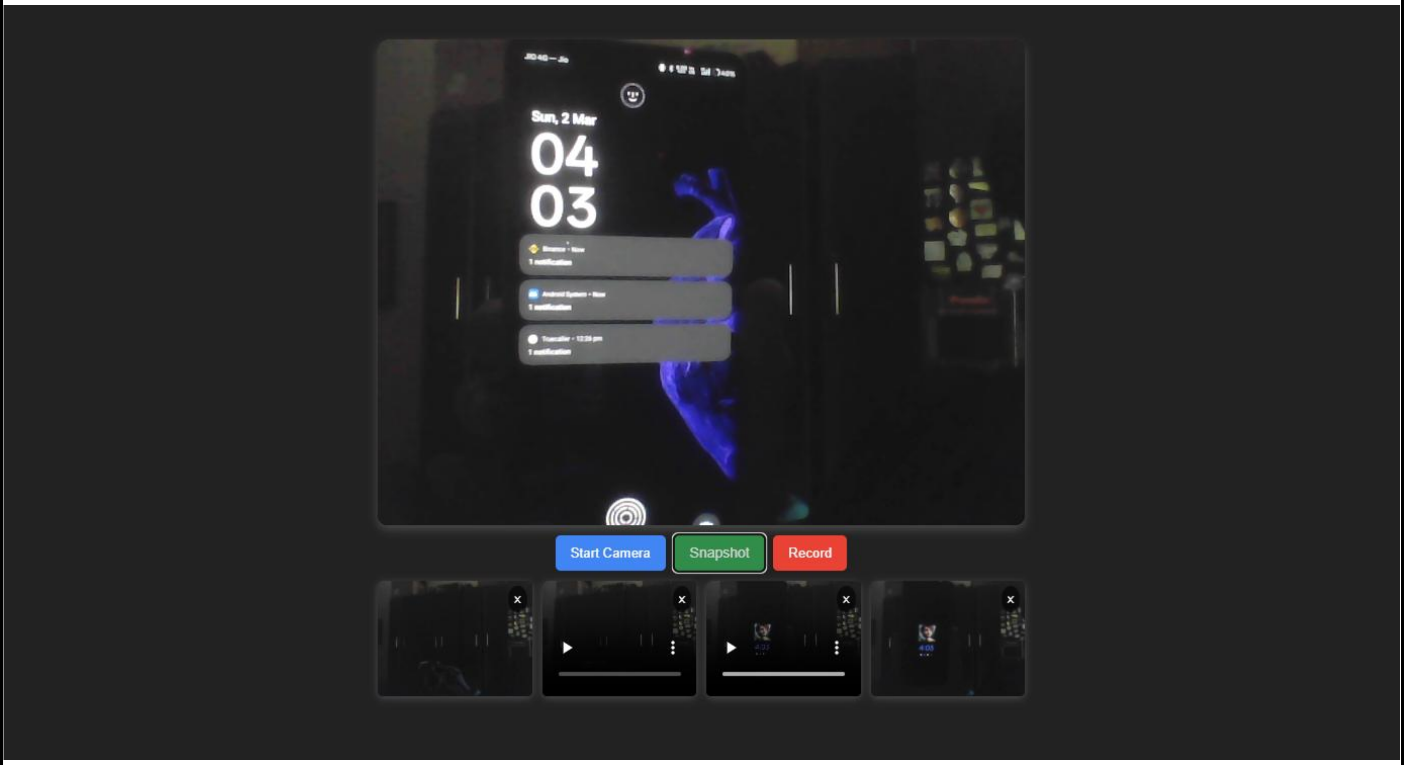
```
</body>
</html>
```



7.Webcam Accessing

https://23bai1214-divyanshu-exercise-6-t-7.netlify.app/

```
<!-- 7. Webcam Accessing  -->

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Webcam Capture</title>
    <style>
        * { margin: 0; padding: 0; box-sizing: border-box; font-family: sans-serif; }
        body { display: flex; flex-direction: column; align-items: center; justify-con-
tent: center; height: 100vh; background: #222; color: white; }
        video, canvas { width: 100%; max-width: 640px; border-radius: 10px; background:
#000; box-shadow: 0 4px 10px rgba(255,255,255,0.2); }
        .controls { display: flex; gap: 10px; margin-top: 10px; }
        button { padding: 10px 15px; border: none; cursor: pointer; border-radius: 5px;
color: white; transition: 0.3s; }
        .primary { background: #4285f4; } .success { background: #34a853; } .danger {
background: #ea4335; }
        button:hover { opacity: 0.8; transform: scale(1.05); }
        .gallery { display: grid; gap: 10px; margin-top: 10px; grid-template-columns: re-
peat(auto-fit, minmax(150px, 1fr)); }
        .gallery div { position: relative; }
        .gallery img, .gallery video { width: 100%; border-radius: 5px; box-shadow: 0 2px
8px rgba(255,255,255,0.2); }
        .delete { position: absolute; top: 5px; right: 5px; background: rgba(0,0,0,0.6);
border: none; color: white; font-size: 14px; padding: 5px; cursor: pointer; border-radius:
50%; }
    </style>
</head>
<body>
    <video id="webcam" autoplay playsinline></video>
    <div class="controls">
        <button id="startBtn" class="primary">Start Camera</button>
        <button id="snapBtn" class="success" disabled>Snapshot</button>
        <button id="recBtn" class="danger" disabled>Record</button>
    </div>
    <div class="gallery" id="gallery"></div>
    <script>
        const video = document.getElementById('webcam'), snapBtn = document.getEle-
mentById('snapBtn'), recBtn = document.getElementById('recBtn'), gallery = document.getEl-
ementById('gallery');
        let stream, recorder, chunks = [], recording = false;

        document.getElementById('startBtn').onclick = async () => {
            if (stream) return stream.getTracks().forEach(t => t.stop()), video.srcObject
= null, stream = null, snapBtn.disabled = recBtn.disabled = true;
            stream = await navigator.mediaDevices.getUserMedia({ video: true, audio: true
});
            video.srcObject = stream; snapBtn.disabled = recBtn.disabled = false;
        };

        snapBtn.onclick = () => {
            let c = document.createElement('canvas'), ctx = c.getContext('2d');
            c.width = video.videoWidth; c.height = video.videoHeight;
            ctx.drawImage(video, 0, 0, c.width, c.height);
            let img = document.createElement('img');
            img.src = c.toDataURL('image/png');
            addToGallery(img);
        };

        recBtn.onclick = () => {
            if (recording) return recorder.stop(), recBtn.textContent = 'Record', record-
ing = false;
            chunks = [];
            recorder = new MediaRecorder(stream);
            recorder.ondataavailable = e => chunks.push(e.data);
            recorder.onstop = () => {
                let vid = document.createElement('video'); vid.src = URL.createObjec-
tURL(new Blob(chunks, { type: 'video/webm' }));
                vid.controls = true;
                addToGallery(vid);
            };
            recorder.start(); recBtn.textContent = 'Stop'; recording = true;
        };

        function addToGallery(media) {
            let wrapper = document.createElement('div'), delBtn = document.createEl-
ement('button');
            delBtn.textContent = '×'; delBtn.className = 'delete';
            delBtn.onclick = () => wrapper.remove();
            wrapper.appendChild(media); wrapper.appendChild(delBtn);
            gallery.appendChild(wrapper);
        }
    </script>
</body>
</html>
```



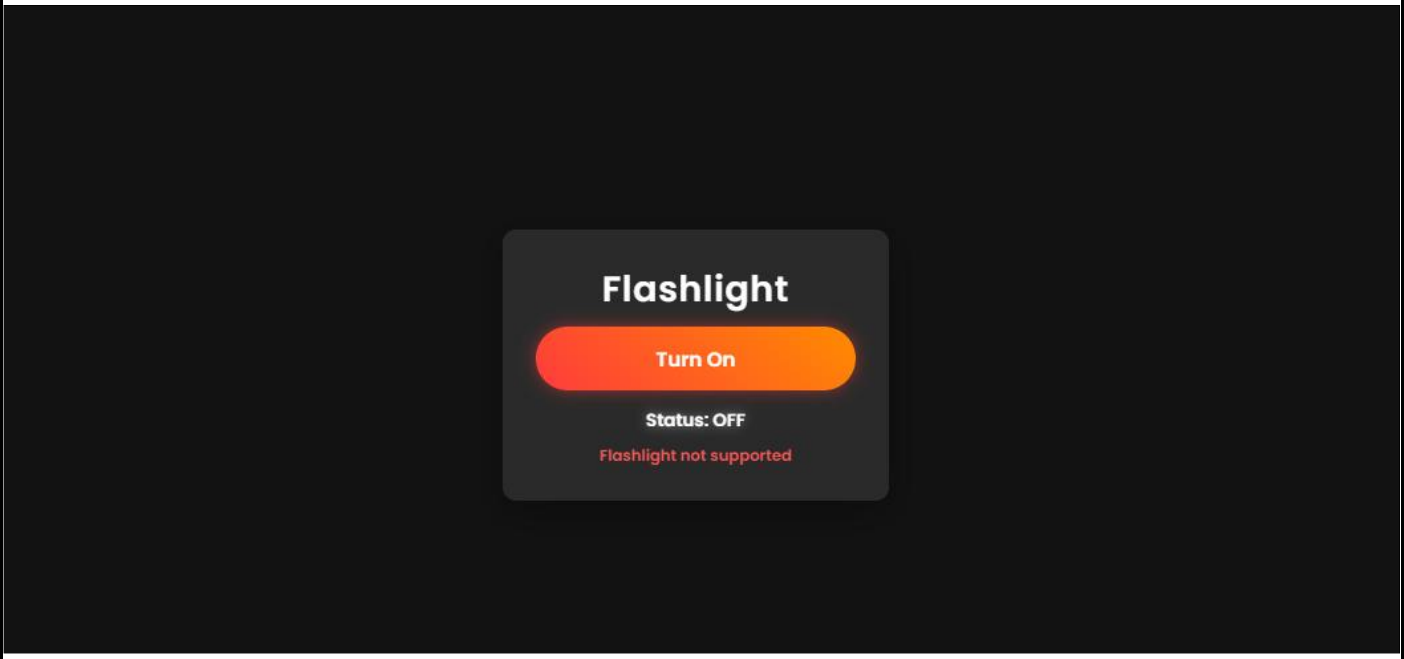
8. Mobile Flashlight

https://23bai1214-divyanshu-exercise-6-t-8.netlify.app/

```
<!-- 8. Mobile Flashlight  -->

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Flashlight</title>
    <style>
        @import url('https://fonts.googleapis.com/css2?family=Poppins:wght@600&dis-
play=swap');
        * { margin: 0; padding: 0; box-sizing: border-box; font-family: 'Poppins', sans-
serif; }
        body { display: flex; align-items: center; justify-content: center; height: 100vh;
background: #121212; color: white; }
```

```html
        .container { text-align: center; padding: 30px; border-radius: 12px; background: rgba(255, 255, 255, 0.1); box-shadow: 0 8px 32px rgba(0, 0, 0, 0.3); backdrop-filter: blur(10px); width: 90%; max-width: 350px; }
        h1 { margin-bottom: 10px; letter-spacing: 1px; }
        #toggleButton { width: 100%; padding: 15px; font-size: 18px; font-weight: bold; cursor: pointer; border-radius: 50px; border: none; transition: 0.3s; background: linear-gradient(45deg, #ff3c3c, #ff8c00); color: white; box-shadow: 0 0 15px rgba(255, 60, 60, 0.5); }
        #toggleButton:active { transform: scale(0.95); }
        #toggleButton.on { background: linear-gradient(45deg, #00ff7f, #00c853); box-shadow: 0 0 20px rgba(0, 255, 127, 0.7); }
        #status { margin-top: 15px; font-size: 16px; text-shadow: 0 0 10px rgba(255, 255, 255, 0.5); }
        .error { margin-top: 10px; font-size: 14px; color: #ff5757; }
    </style>
</head>
<body>
    <div class="container">
        <h1>Flashlight</h1>
        <button id="toggleButton">Turn On</button>
        <div id="status">Status: OFF</div>
        <div id="error" class="error"></div>
        <video id="video" autoplay playsinline style="display:none;"></video>
    </div>
    <script>
        const btn = document.getElementById('toggleButton'), statusText = document.getElementById('status'), errorText = document.getElementById('error'), video = document.getElementById('video');
        let stream, track, flashlightOn = false;
        btn.onclick = async () => {
            try {
                if (!flashlightOn) {
                    stream = await navigator.mediaDevices.getUserMedia({ video: { facingMode: 'environment', advanced: [{ torch: true }] } });
                    video.srcObject = stream;
                    track = stream.getVideoTracks()[0];
                    await track.applyConstraints({ advanced: [{ torch: true }] });
                    flashlightOn = true;
                    updateUI();
                } else {
                    track.stop();
                    flashlightOn = false;
                    updateUI();
                }
            } catch (err) {
                errorText.textContent = 'Flashlight not supported';
            }
        };
        function updateUI() {
            btn.textContent = flashlightOn ? 'Turn Off' : 'Turn On';
            btn.classList.toggle('on', flashlightOn);
            statusText.textContent = `Status: ${flashlightOn ? 'ON' : 'OFF'}`;
        }
    </script>
```

```html
</body>
</html>
```



Status: OFF

Flashlight not supported