**SOUL FITNESS TRACKER CODING**

**FRONT PAGE:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>soul fitness tracker</title>

<style>

/\* Global Styles \*/

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

font-family: 'palatino', serif;

font-weight: bold;

font-style: italic;

}

/\* Background Styling \*/

body {

background: url('front animi.jpg') no-repeat center center fixed;

background-size: cover;

display: flex;

flex-direction: column;

justify-content: flex-start;

align-items: center;

height: 100vh;

text-align: center;

color: black;

padding-top: 25vh;

position: relative;

}

/\* Heading Styles \*/

h1 {

font-size: 60px;

text-transform: uppercase;

letter-spacing: 3px;

text-shadow: 0 0 10px rgba(0, 0, 0, 0.8), 0 0 20px rgba(0, 0, 0, 0.6);

margin-bottom: 20px;

}

/\* Neon Black Button \*/

.btn {

display: inline-block;

font-size: 22px;

padding: 12px 30px;

background: black;

color: white;

border: 2px solid black;

cursor: pointer;

border-radius: 50px;

text-decoration: none;

transition: 0.3s ease-in-out;

font-weight: bold;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.8), 0 0 20px rgba(0, 0, 0, 0.6);

}

.btn:hover {

background: white;

color: black;

box-shadow: 0 0 20px rgba(0, 0, 0, 0.8), 0 0 40px rgba(0, 0, 0, 0.6);

}

/\* Neon Black Logout Button \*/

.logout-btn {

position: absolute;

top: 20px;

right: 20px;

font-size: 18px;

padding: 10px 20px;

background: black;

color: white;

border: 2px solid black;

cursor: pointer;

border-radius: 20px;

text-decoration: none;

transition: background 0.3s ease-in-out, transform 0.3s ease-in-out, box-shadow 0.3s ease-in-out;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.8), 0 0 20px rgba(0, 0, 0, 0.6);

}

.logout-btn:hover {

background: white;

color: black;

transform: scale(1.1);

box-shadow: 0 0 20px rgba(0, 0, 0, 0.8), 0 0 40px rgba(0, 0, 0, 0.6);

}

</style>

</head>

<body>

<a href="loginjava.html" class="logout-btn">Logout</a>

<div class="content">

<h1>soul fitness tracker</h1>

<h2>sweat, smile, shine<h2><br>

<a href="loginjava.html" class="btn">Get Started</a>

</div>

</body>

</html>

**SIGNUP PAGE:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>soul fitness tracker- Register</title>

<!-- FontAwesome for Icons -->

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.2/css/all.min.css">

<style>

/\* General Styling \*/

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

font-family: 'Poppins', sans-serif;

}

/\* Background Styling \*/

body {

background: url('login animi.jpg') no-repeat center center/cover;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

color: white;

}

/\* Form Container \*/

.main {

background: rgba(0, 0, 0, 0.8);

padding: 30px;

border-radius: 15px;

box-shadow: 0px 5px 15px rgba(0, 0, 0, 0.3);

width: 350px;

text-align: center;

animation: fadeIn 1.5s ease-in-out;

}

h2 {

color: #4CAF50;

margin-bottom: 20px;

font-size: 28px;

}

label {

display: block;

margin-top: 10px;

text-align: left;

font-size: 14px;

color: #ddd;

}

input, select {

width: 100%;

padding: 10px;

margin-top: 5px;

border: none;

border-radius: 5px;

outline: none;

background: rgba(255, 255, 255, 0.1);

color: white;

font-size: 16px;

}

input::placeholder {

color: #ccc;

}

select option {

color: black;

background: white;

}

/\* Password Container \*/

.password-container {

position: relative;

}

.eye-icon {

position: absolute;

right: 15px;

top: 50%;

transform: translateY(-50%);

cursor: pointer;

color: #bbb;

font-size: 18px;

}

.eye-icon:hover {

color: #4CAF50;

}

/\* Button Styling \*/

button {

width: 100%;

padding: 12px;

margin-top: 20px;

border: none;

border-radius: 10px;

background-color: #4CAF50;

color: white;

font-size: 18px;

font-weight: bold;

cursor: pointer;

transition: all 0.3s ease-in-out;

}

button:hover {

background-color: #45a049;

transform: scale(1.05);

}

/\* Fade In Animation \*/

@keyframes fadeIn {

from {

opacity: 0;

transform: translateY(-20px);

}

to {

opacity: 1;

transform: translateY(0);

}

}

/\* Responsive Design \*/

@media (max-width: 768px) {

.main {

width: 90%;

}

}

</style>

</head>

<body>

<div class="main">

<h2>SIGN UP</h2>

<form id="sign-up-form" onsubmit="redirectToDashboard(event)">

<label for="first">First Name:</label>

<input type="text" id="first" name="first" placeholder="John" required>

<label for="last">Last Name:</label>

<input type="text" id="last" name="last" placeholder="Doe" required>

<label for="email">Email:</label>

<input type="email" id="email" name="email" placeholder="john@example.com" required>

<label for="password">Password:</label>

<div class="password-container">

<input type="password" id="password" name="password" required>

<i class="eye-icon fas fa-eye" onclick="togglePassword('password', this)"></i>

</div>

<label for="repassword">Confirm Password:</label>

<div class="password-container">

<input type="password" id="repassword" name="repassword" required>

<i class="eye-icon fas fa-eye" onclick="togglePassword('repassword', this)"></i>

</div>

<label for="mobile">Contact:</label>

<input type="text" id="mobile" name="mobile" maxlength="10" placeholder="1234567890" required>

<label for="gender">Gender:</label>

<select id="gender" name="gender" required>

<option value="" disabled selected>Select</option>

<option value="male">Male</option>

<option value="female">Female</option>

<option value="other">Other</option>

</select>

<button type="submit">LET'S START</button>

<p style="margin-top: 15px; color: #ddd;">

Already have an account?

<a href="alreadylogin.html">Login</a>

</p>

</form>

</div>

<script>

function togglePassword(inputId, icon) {

const passwordField = document.getElementById(inputId);

if (passwordField.type === "password") {

passwordField.type = "text";

icon.classList.remove("fa-eye");

icon.classList.add("fa-eye-slash");

} else {

passwordField.type = "password";

icon.classList.remove("fa-eye-slash");

icon.classList.add("fa-eye");

}

}

function redirectToDashboard(event) {

event.preventDefault(); // Prevents actual form submission

window.location.href = "DASHBOARD 2X2.html"; // Redirects to the dashboard

}

</script>

</body>

</html>

**LOGIN PAGE:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>soul fitness tracker- Login</title>

<!-- FontAwesome for Icons -->

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.2/css/all.min.css">

<style>

/\* General Styling \*/

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

font-family: 'Poppins', sans-serif;

}

/\* Background Styling \*/

body {

background: url('already.jpg') no-repeat center center/cover;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

color: white;

}

/\* Form Container \*/

.main {

background: rgba(0, 0, 0, 0.8);

padding: 30px;

border-radius: 15px;

box-shadow: 0px 5px 15px rgba(0, 0, 0, 0.3);

width: 350px;

text-align: center;

animation: fadeIn 1.5s ease-in-out;

}

h2 {

color: #4CAF50;

margin-bottom: 20px;

font-size: 28px;

}

label {

display: block;

margin-top: 10px;

text-align: left;

font-size: 14px;

color: #ddd;

}

input {

width: 100%;

padding: 10px;

margin-top: 5px;

border: none;

border-radius: 5px;

outline: none;

background: rgba(255, 255, 255, 0.1);

color: white;

font-size: 16px;

}

input::placeholder {

color: #ccc;

}

/\* Password Container \*/

.password-container {

position: relative;

}

.eye-icon {

position: absolute;

right: 15px;

top: 50%;

transform: translateY(-50%);

cursor: pointer;

color: #bbb;

font-size: 18px;

}

.eye-icon:hover {

color: #4CAF50;

}

/\* Button Styling \*/

button {

width: 100%;

padding: 12px;

margin-top: 20px;

border: none;

border-radius: 10px;

background-color: #4CAF50;

color: white;

font-size: 18px;

font-weight: bold;

cursor: pointer;

transition: all 0.3s ease-in-out;

}

button:hover {

background-color: #45a049;

transform: scale(1.05);

}

/\* Fade In Animation \*/

@keyframes fadeIn {

from {

opacity: 0;

transform: translateY(-20px);

}

to {

opacity: 1;

transform: translateY(0);

}

}

/\* Responsive Design \*/

@media (max-width: 768px) {

.main {

width: 90%;

}

}

</style>

</head>

<body>

<div class="main">

<h2>LOGIN</h2>

<form id="login-form">

<label for="email">Email:</label>

<input type="email" id="email" name="email" placeholder="john@example.com" required>

<label for="password">Password:</label>

<div class="password-container">

<input type="password" id="password" name="password" required>

<i class="eye-icon fas fa-eye" onclick="togglePassword('password', this)"></i>

</div>

<button type="submit">LOGIN</button>

<p style="margin-top: 15px; color: #ddd;">

Don't have an account?

<a href="loginjava.html">Sign Up</a>

</p>

</form>

</div>

<script>

function togglePassword(inputId, icon) {

const passwordField = document.getElementById(inputId);

if (passwordField.type === "password") {

passwordField.type = "text";

icon.classList.remove("fa-eye");

icon.classList.add("fa-eye-slash");

} else {

passwordField.type = "password";

icon.classList.remove("fa-eye-slash");

icon.classList.add("fa-eye");

}

}

document.addEventListener("DOMContentLoaded", function () {

const form = document.getElementById("login-form");

const email = document.getElementById("email");

const password = document.getElementById("password");

form.addEventListener("submit", function (event) {

event.preventDefault(); // Prevent form submission

// Validate email format

if (!validateEmail(email.value)) {

alert("Please enter a valid email address!");

return;

}

// Validate password (not empty)

if (password.value.trim() === "") {

alert("Please enter your password!");

return;

}

alert("Login Successful!");

window.location.href = "DASHBOARD 2x2.html"; // Redirect to dashboard

});

function validateEmail(email) {

const emailPattern = /^[a-zA-Z0-9.\_%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$/;

return emailPattern.test(email);

}

});

</script>

</body>

</html>

**DASHBROAD:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Fitness Dashboard</title>

<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.min.css" rel="stylesheet">

<style>

@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;600&display=swap');

body {

font-family: 'Poppins', sans-serif;

background: url('move.jpg') no-repeat center center/cover;

color: #ffffff;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

margin: 0;

flex-direction: column;

padding: 10px;

}

.back-button {

position: absolute;

top: 20px;

left: 20px;

background: black;

color: #00ffcc;

padding: 10px 20px;

text-decoration: none;

border-radius: 8px;

font-weight: bold;

font-size: 16px;

box-shadow: 0 0 15px #00ffcc;

transition: 0.3s ease-in-out;

border: 2px solid #00ffcc;

}

.back-button:hover {

box-shadow: 0 0 25px #00ffcc;

color: #ffffff;

}

.link-button {

margin-top: 15px;

display: inline-block;

background: black;

color: #00ffcc;

padding: 12px 25px;

text-decoration: none;

border-radius: 8px;

font-weight: bold;

font-size: 16px;

box-shadow: 0 0 15px #00ffcc;

transition: 0.3s ease-in-out;

border: 2px solid #00ffcc;

}

.link-button:hover {

background: black;

box-shadow: 0 0 25px #00ffcc;

color: #ffffff;

}

.container {

text-align: center;

background: rgba(30, 30, 30, 0.8);

padding: 15px;

border-radius: 10px;

box-shadow: 0px 0px 10px #00ffcc;

margin: 10px 0;

border: 1.5px solid #00ffcc;

}

.dashboard-container {

display: grid;

grid-template-columns: repeat(2, 1fr);

grid-gap: 10px;

max-width: 600px;

width: 100%;

padding: 5px;

}

.tracker {

background: rgba(40, 40, 40, 0.85);

backdrop-filter: blur(8px);

padding: 15px;

border-radius: 8px;

box-shadow: 0 0 8px rgba(0, 255, 255, 0.5);

text-align: center;

width: 100%;

height: 200px;

display: flex;

flex-direction: column;

justify-content: space-between;

}

.tracker.calories-tracker {

height: 300px;

}

.graphs-container {

display: flex;

justify-content: space-between;

margin-top: 20px;

width: 100%;

}

.graph-container {

width: 48%;

background: rgba(30, 30, 30, 0.9);

padding: 20px;

border-radius: 8px;

box-shadow: 0 0 8px rgba(0, 255, 255, 0.5);

text-align: center;

}

input {

width: 100%;

padding: 6px;

margin: 4px 0;

border: none;

border-radius: 5px;

outline: none;

text-align: center;

background: #333;

color: #fff;

font-size: 12px;

}

button {

background: #00ffff;

border: none;

padding: 6px;

border-radius: 5px;

cursor: pointer;

transition: 0.3s;

color: #121212;

font-weight: bold;

font-size: 12px;

}

button:hover {

background: #00b3b3;

}

canvas {

max-width: 100%;

height: 100px;

}

</style>

</head>

<body>

<div class="main" style="text-align: center;">

<h2>DASHBOARD TRACKER</h2>

<div class="dashboard-container">

<div class="tracker">

<h4>Workout</h4>

<form>

<input type="text" id="exercise" placeholder="Exercise">

<input type="number" id="duration" placeholder="Duration (mins)">

<button type="button" onclick="addWorkout()">Track</button>

</form>

</div>

<div class="tracker">

<h4>Diet & Water</h4>

<form>

<input type="number" id="caloriesMeal" placeholder="Calories">

<input type="number" id="water" placeholder="Water (ml)">

<button type="button" onclick="addDiet()">Track</button>

</form>

</div>

<div class="tracker calories-tracker">

<h4>Calories</h4>

<form>

<input type="number" id="age" placeholder="Age">

<input type="number" id="height" placeholder="Height (cm)">

<input type="number" id="weight" placeholder="Weight (kg)">

<button type="button" onclick="calculateCalories()">Track</button>

</form>

<p>Calories: <span id="caloriesResult">--</span> kcal</p>

</div>

<div class="tracker">

<h4>Heartbeat</h4>

<button onclick="trackHeartbeat()">Track</button>

<p>Heart Rate: <span id="bpm">--</span> BPM</p>

</div>

</div>

<div class="graphs-container">

<div class="graph-container">

<h4>Workout Progress</h4>

<canvas id="workoutChart"></canvas>

</div>

<div class="graph-container">

<h4>Diet & Water Intake</h4>

<canvas id="dietChart"></canvas>

</div>

</div>

<script>

function trackHeartbeat() {

document.getElementById("bpm").innerText = Math.floor(Math.random() \* (120 - 60) + 60);

}

function calculateCalories() {

let age = document.getElementById("age").value;

let height = document.getElementById("height").value;

let weight = document.getElementById("weight").value;

if (age && height && weight) {

let calories = Math.round((10 \* weight) + (6.25 \* height) - (5 \* age) + 5);

document.getElementById("caloriesResult").innerText = calories;

}

}

function addDiet() {

let calMeal = document.getElementById("caloriesMeal").value;

let water = document.getElementById("water").value;

if (calMeal && water) {

updateChart("dietChart", [parseInt(water)], "Water Intake (ml)", "rgba(0, 255, 255, 0.5)");

}

}

function addWorkout() {

let duration = document.getElementById("duration").value;

if (duration) {

updateChart("workoutChart", [parseInt(duration)], "Workout Duration (mins)", "rgba(0, 255, 0, 0.5)");

}

}

function updateChart(id, data, label, color) {

new Chart(document.getElementById(id).getContext("2d"), {

type: "bar",

data: {

labels: Array.from({ length: data.length }, (\_, i) => i + 1),

datasets: [{ label: label, data: data, backgroundColor: color }]

},

options: { scales: { y: { beginAtZero: true } } }

});

}

</script>

</body><br>

<body>

<a href="loginjava.html" class="back-button">&#8592; Back</a>

</div>

<!-- Button to navigate to Blood Pressure Tracker -->

<a href="bp tracker.html" class="link-button">Go to BP Tracker</a>

</div><br>

<div class="content">

<button class="MOVE TO CALENDAR">

<a href="calendar store.html" class="btn"> 📅 MOVE TO CALENDAR</a>

</div><br>

<div class="content">

<button class="MOVE TO TIMER & STOPWATCH">

<a href="lol.html" class="btn"> MOVE TO 🕑 TIMER & STOPWATCH</a>

</div>

<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

<script src="dashboard.js"></script>

</body>

</html>

**BLOOD PRESSURE TRACKER:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Blood Pressure Tracker</title>

<style>

body {

font-family: palatino, serif;

text-align: center;

background:url('bp im.jpg') no-repeat center center/cover;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

margin: 0

padding: 20px;

}

.container {

max-width: 400px;

margin: auto;

background: #0d0d0d;

color: #fff;

padding: 20px;

border-radius: 10px;

box-shadow: 0px 0px 10px rgba(0, 255, 255, 0.5);

}

input {

width: 95%;

padding: 10px;

margin: 10px 0;

border: 1px solid #0ff;

border-radius: 5px;

background: #ffffff;

color: #000;

}

button {

background: #ffffff;

color: black;

border: none;

padding: 10px;

width: 100%;

cursor: pointer;

border-radius: 5px;

}

button:hover {

background: #e0e0e0;

}

.result {

margin-top: 20px;

font-size: 18px;

font-weight: bold;

}

.back-button {

position: absolute;

top: 20px;

left: 20px;

background: #0d0d0d;

color: #0ff;

padding: 10px 15px;

border-radius: 5px;

text-decoration: none;

box-shadow: 0px 0px 10px rgba(0, 255, 255, 0.7);

font-weight: bold;

}

.back-button:hover {

background: #1a1a1a;

}

</style>

</head>

<body>

<a href="javascript:history.back()" class="back-button">&#8592; Back</a>

<div class="container">

<h2>Blood Pressure Tracker</h2>

<label for="systolic">Systolic (Top Number)</label>

<input type="number" id="systolic" placeholder="Enter systolic value">

<label for="diastolic">Diastolic (Bottom Number)</label>

<input type="number" id="diastolic" placeholder="Enter diastolic value">

<button onclick="checkBP()">tracker</button>

<div class="result" id="result"></div>

</div>

<script>

function checkBP() {

let systolic = parseInt(document.getElementById('systolic').value);

let diastolic = parseInt(document.getElementById('diastolic').value);

let result = document.getElementById('result');

if (isNaN(systolic) || isNaN(diastolic)) {

result.innerHTML = "Please enter valid numbers for both values.";

result.style.color = "red";

return;

}

let category = "";

let color = "white";

if (systolic < 120 && diastolic < 80) {

category = "Normal";

color = "green";

} else if (systolic >= 120 && systolic <= 129 && diastolic < 80) {

category = "Elevated";

color = "orange";

} else if ((systolic >= 130 && systolic <= 139) || (diastolic >= 80 && diastolic <= 89)) {

category = "High Blood Pressure (Stage 1)";

color = "red";

} else if (systolic >= 140 || diastolic >= 90) {

category = "High Blood Pressure (Stage 2)";

color = "darkred";

}

if (systolic >= 180 || diastolic >= 120) {

category = "Hypertensive Crisis! Seek medical attention immediately.";

color = "purple";

}

result.innerHTML = "Your Blood Pressure Category: " + category;

result.style.color = color;

}

</script>

</body>

</html>

**CALENDER STORE:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Calendar and Workout Tracker</title>

<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

<style>

body {

font-family: 'Poppins', sans-serif;

display: flex;

flex-direction: column;

align-items: center;

height: 100vh;

background: url('calendar animi.jpg') no-repeat center center/cover;

color: white;

padding: 10px;

}

.back-button {

position: absolute;

top: 10px;

left: 10px;

background: black;

color: #00ffcc;

padding: 10px 15px;

border-radius: 5px;

text-decoration: none;

font-weight: bold;

box-shadow: 0 0 10px #00ffcc;

transition: 0.3s;

}

.back-button:hover {

box-shadow: 0 0 15px #00ffcc;

}

.container {

display: flex;

gap: 30px;

flex-wrap: wrap;

justify-content: center;

margin-top: 10px;

width: 75%;

}

.calendar-container, .todo-container {

background: rgba(0, 0, 0, 0.9);

padding: 15px;

border-radius: 10px;

box-shadow: 0 0 10px #00ffcc;

text-align: center;

width: 70%;

transition: 0.3s;

}

.calendar-container:hover, .todo-container:hover {

box-shadow: 0 0 15px #00ffcc;

}

.chart-container {

display: block;

margin-top: 20px;

width: 40%;

background: black;

padding: 15px;

border-radius: 10px;

box-shadow: 0 0 10px #ff00ff;

text-align: center;

transition: 0.3s;

}

canvas {

max-width: 300px;

max-height: 300px;

margin: auto;

}

</style>

</head>

<body>

<a href="#" class="back-button" onclick="history.back()">⬅ Back</a>

<h1>📅 CALENDAR & 🏋️ WORKOUT TRACKER</h1>

<div class="container">

<div class="calendar-container">

<h3>📅 Calendar</h3>

<input type="date" id="workoutDate" class="calendar">

</div>

<div class="todo-container">

<h3>🏋️ Completed Workouts</h3>

<input type="text" id="workoutInput" placeholder="Add a workout">

<button onclick="addWorkout()">Add</button>

<ul id="workoutList"></ul>

</div>

</div>

<div class="chart-container" id="chartContainer">

<h3>📊 Workout Overview</h3>

<canvas id="workoutChart"></canvas>

</div>

<script>

let workoutData = JSON.parse(localStorage.getItem('workouts')) || {};

let colors = JSON.parse(localStorage.getItem('workoutColors')) || {};

let colorPalette = ["#FFD700", "#36A2EB", "#FF6384", "#4BC0C0", "#9966FF", "#FF9F40"];

let colorIndex = Object.keys(colors).length;

function addWorkout() {

let input = document.getElementById("workoutInput");

let dateInput = document.getElementById("workoutDate").value;

let workout = input.value.trim();

let dateTime = dateInput ? `${dateInput} ${new Date().toLocaleTimeString()}` : new Date().toLocaleString();

if (workout === "") return;

let ul = document.getElementById("workoutList");

let li = document.createElement("li");

li.innerHTML = `${workout} <span style='font-size: 0.8em; color: gray;'>(${dateTime})</span> <span class='delete' onclick='removeWorkout(this, "${workout}", "${dateTime}")'>✖</span>`;

ul.appendChild(li);

if (!Array.isArray(workoutData[workout])) {

workoutData[workout] = [];

}

workoutData[workout].push(dateTime);

if (!colors[workout]) {

colors[workout] = colorPalette[colorIndex % colorPalette.length];

colorIndex++;

}

localStorage.setItem('workouts', JSON.stringify(workoutData));

localStorage.setItem('workoutColors', JSON.stringify(colors));

updateChart();

input.value = "";

}

function removeWorkout(element, workout, dateTime) {

element.parentElement.remove();

if (Array.isArray(workoutData[workout])) {

workoutData[workout] = workoutData[workout].filter(entry => entry !== dateTime);

if (workoutData[workout].length === 0) delete workoutData[workout];

}

localStorage.setItem('workouts', JSON.stringify(workoutData));

updateChart();

}

function updateChart() {

let ctx = document.getElementById("workoutChart").getContext("2d");

if (window.workoutChartInstance) window.workoutChartInstance.destroy();

window.workoutChartInstance = new Chart(ctx, {

type: "pie",

data: {

labels: Object.keys(workoutData),

datasets: [{

data: Object.keys(workoutData).map(workout => workoutData[workout].length),

backgroundColor: Object.keys(workoutData).map(workout => colors[workout] || "#ddd"),

}],

},

options: {

plugins: {

legend: {

position: 'right',

labels: {

color: 'white',

}

}

}

}

});

}

function loadWorkouts() {

let ul = document.getElementById("workoutList");

for (let workout in workoutData) {

if (Array.isArray(workoutData[workout])) {

workoutData[workout].forEach(dateTime => {

let li = document.createElement("li");

li.innerHTML = `${workout} <span style='font-size: 0.8em; color: gray;'>(${dateTime})</span> <span class='delete' onclick='removeWorkout(this, "${workout}", "${dateTime}")'>✖</span>`;

ul.appendChild(li);

});

}

}

updateChart();

}

window.onload = loadWorkouts;

</script>

</body>

</html>

CLOCK & WORKOUT:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Clock & Workout</title>

<style>

@import url('https://fonts.googleapis.com/css2?family=Rajdhani:wght@400;700&display=swap');

body {

font-family: 'Rajdhani', sans-serif;

display: flex;

justify-content: space-between;

align-items: center;

height: 100vh;

background: url('sea animi.jpg') no-repeat center center fixed;

background-size: cover;

color: white;

margin: 0;

padding: 20px;

animation: fadeIn 1.5s ease-in-out;

}

.back-button {

position: absolute;

top: 20px;

left: 20px;

background: black;

color: #00ffcc;

padding: 10px 20px;

text-decoration: none;

border-radius: 8px;

font-weight: bold;

font-size: 16px;

box-shadow: 0 0 15px #00ffcc;

transition: 0.3s ease-in-out;

border: 2px solid #00ffcc;

}

.back-button:hover {

box-shadow: 0 0 25px #00ffcc;

color: #ffffff;

}

@keyframes fadeIn {

from { opacity: 0; }

to { opacity: 1; }

}

.left-container, .right-container {

width: 48%;

height: 90vh;

padding: 20px;

background:transparent /\* rgba(30, 30, 30, 0.4);\*/

border-radius: 10px;

box-shadow: 0px 0px 10px cyan;

overflow: auto;

}

.left-container {

animation: glowEffect 3s infinite alternate;

}

@keyframes glowEffect {

from { box-shadow: 0px 0px 10px cyan; }

to { box-shadow: 0px 0px 20px cyan; }

}

.container {

text-align: center;

background:/\*transparent\*/ rgba(30, 30, 30, 0.4);

padding: 15px;

border-radius: 10px;

box-shadow: 0px 0px 10px #00ffcc;

margin: 10px 0;

border: 1.5px solid #00ffcc;

}

.time-display {

font-size: 1.9em;

margin: 10px 0;

font-weight: bold;

color: #00ffcc;

}

button {

margin: 3px;

padding: 8px 15px;

border: none;

background: #00ffcc;

color: #121212;

cursor: pointer;

border-radius: 6px;

font-size: 0.9em;

font-weight: bold;

transition: transform 0.3s;

}

button:hover {

transform: scale(1.1);

}

.exercise-container {

display: grid;

grid-template-columns: repeat(3, 1fr);

gap: 15px;

justify-content: center;

}

.modal-content img {

width: 300px; /\* Adjust as needed \*/

height: 300px; /\* Adjust as needed \*/

object-fit: cover; /\* Ensures the image scales properly \*/

border-radius: 8px;

}

.exercise {

width: 75px;

cursor: pointer;

text-align: center;

padding: 6px;

background: rgba(30, 30, 30, 0.4);

border-radius: 6px;

border: 2px solid cyan;

box-shadow: 0 0 6px cyan;

transition: transform 0.3s, box-shadow 0.3s;

}

.exercise:hover {

transform: scale(1.1);

box-shadow: 0 0 15px cyan;

}

.exercise img {

width: 100%;

border-radius: 4px;

}

.modal {

display: none;

position: fixed;

top: 0;

left: 0;

width: 100%;

height: 100%;

background: rgba(30, 30, 30, 0.4); /\* Dark overlay \*/

justify-content: center;

align-items: center;

}

.modal-content {

position: relative;

text-align: center;

background: transparent;

border: none;

}

.modal-content img {

width: 80%;

max-width: 600px;

height: auto;

border-radius: 8px;

}

</style>

</head>

<body>

<a href="frontpage.html" class="back-button">&#8592; Back</a>

<div class="left-container">

<h1 style="text-align:center;">Time Breaker 🕑</h1>

<div class="container">

<h2>Clock</h2>

<div id="clock" class="time-display"></div>

</div>

<div class="container">

<h2>Timer</h2>

<div id="timer" class="time-display">00:00</div>

<button onclick="startTimer()">Start</button>

<button onclick="pauseTimer()">Pause</button>

<button onclick="resetTimer()">Reset</button>

</div>

<div class="container">

<h2>Stopwatch</h2>

<div id="stopwatch" class="time-display">00:00:00</div>

<button onclick="startStopwatch()">Start</button>

<button onclick="pauseStopwatch()">Pause</button>

<button onclick="resetStopwatch()">Reset</button>

</div>

</div>

<div class="right-container">

<h1 style="text-align:center;">Basic Warm-up Exercises</h1>

<div class="exercise-container">

<body>

<div class="exercise" onclick="openModal('Dumbbell Lateral Lunge', 'Works legs and glutes, improves balance.', '30 seconds each side', 'lubge animi.jpg')">

<img src="lubge animi.jpg" alt="Dumbbell Lateral Lunge">

<p>Dumbbell Lateral Lunge</p>

</div>

<div class="exercise" onclick="openModal('Seated Hamstring Stretch', 'Stretches the hamstrings and lower back.', '20-30 seconds each leg', 'seated stretch.jpg')">

<img src="seated stretch.jpg" alt="Seated Hamstring Stretch">

<p>Seated Hamstring Stretch</p>

</div>

<div class="exercise" onclick="openModal('Plank', 'Strengthens core and improves stability.', 'Hold for 30-60 seconds', 'plank animi.jpg')">

<img src="plank animi.jpg" alt="Plank">

<p>Plank</p>

</div>

<div class="exercise" onclick="openModal('Skipping', 'Improves coordination and cardiovascular endurance.', '1-2 minutes', 'Skipping animi.jpg')">

<img src="Skipping animi.jpg" alt="Skipping">

<p>Skipping</p>

</div>

<div class="exercise" onclick="openModal('Squat', 'Strengthens lower body muscles and improves flexibility.', '12-15 reps', 'squat animi.jpg')">

<img src="squat animi.jpg" alt="Squat">

<p>Squat</p>

</div>

<div class="exercise" onclick="openModal('Standing Quadriceps Stretch', 'Stretches the quadriceps and improves flexibility.', '20-30 seconds each leg', 'stretch animi.jpg')">

<img src="stretch animi.jpg" alt="Standing Quadriceps Stretch">

<p>Standing Quadriceps Stretch</p>

</div>

</div>

</div>

<div id="modal" class="modal" onclick="closeModal()">

<div class="modal-content">

<span class="close" onclick="closeModal()">&times;</span>

<h2 id="exercise-title"></h2>

<img id="exercise-img" src="" alt="Exercise">

<p id="exercise-definition"></p>

<p id="exercise-time"></p>

</div>

</div>

<script>

function openModal(title, imgSrc) {

document.getElementById("exercise-title").innerText = title;

document.getElementById("exercise-img").src = imgSrc;

document.getElementById("modal").style.display = "flex";

}

function closeModal() {

document.getElementById("modal").style.display = "none";

}

</script>

</body>

</div>

</div>

<script>

function updateClock() {

document.getElementById("clock").innerText = new Date().toLocaleTimeString();

}

setInterval(updateClock, 1000);

updateClock();

let timerInterval, timerTime = 0;

function startTimer() {

if (!timerInterval) {

timerInterval = setInterval(() => {

timerTime++;

document.getElementById("timer").innerText = new Date(timerTime \* 1000).toISOString().substr(14, 5);

}, 1000);

}

}

function pauseTimer() { clearInterval(timerInterval); timerInterval = null; }

function resetTimer() { pauseTimer(); timerTime = 0; document.getElementById("timer").innerText = "00:00"; }

// Stopwatch (Fast speed with step-by-step increments)

let stopwatchInterval;

let stopwatchTime = 0;

function startStopwatch() {

if (!stopwatchInterval) {

stopwatchInterval = setInterval(() => {

stopwatchTime++;

let date = new Date(stopwatchTime \* 1000);

document.getElementById("stopwatch").innerText = date.toISOString().substr(11, 8);

}, 200); // Updates every 200ms for a faster effect but step-by-step

}

}

function pauseStopwatch() { clearInterval(stopwatchInterval); stopwatchInterval = null; }

function resetStopwatch() { pauseStopwatch(); stopwatchTime = 0; document.getElementById("stopwatch").innerText = "00:00:00"; }

</script>

<!-- Link the external JavaScript file at the end of the body -->

<script src="comtral.js"></script>

<script>

function openModal(title, definition, time, imgSrc) {

document.getElementById("exercise-title").innerText = title;

document.getElementById("exercise-definition").innerText = definition;

document.getElementById("exercise-time").innerText = "Duration: " + time;

document.getElementById("exercise-img").src = imgSrc;

document.getElementById("modal").style.display = "flex";

}

function closeModal() {

document.getElementById("modal").style.display = "none";

}

// Prevent accidental closing when clicking inside the modal

document.getElementById("modal").addEventListener("click", function (event) {

if (event.target.classList.contains("modal")) {

closeModal();

}

});

</script>

</body>

</html>

**BACKEND CONDING:**

**SCRIPT1:**

let taskData = JSON.parse(localStorage.getItem("tasks")) || {};

let colors = JSON.parse(localStorage.getItem("taskColors")) || {};

let colorPalette = ["#FFD700", "#36A2EB", "#FF6384", "#4BC0C0", "#9966FF", "#FF9F40", "#8B4513", "#00FF00", "#9400D3", "#FF4500", "#55ff33", "#af33ff"];

let colorIndex = Object.keys(colors).length;

// Load saved tasks on page load

window.onload = function () {

loadTasks();

};

function addTask() {

let input = document.getElementById("taskInput");

let task = input.value.trim();

if (task === "") return;

let date = new Date().toLocaleDateString();

let time = new Date().toLocaleTimeString();

let ul = document.getElementById("taskList");

let li = document.createElement("li");

li.innerHTML = `${task} <span style='font-size: 0.8em; color: gray;'>(${date} - ${time})</span> <span class='delete' onclick='removeTask(this, "${task}")'>✖</span>`;

ul.appendChild(li);

if (!taskData[task]) {

taskData[task] = 1;

colors[task] = colorPalette[colorIndex % colorPalette.length];

colorIndex++;

} else {

taskData[task]++;

}

saveTasks();

document.getElementById("chartContainer").style.display = "block";

updateChart();

input.value = "";

}

function removeTask(element, task) {

element.parentElement.remove();

if (taskData[task]) {

taskData[task]--;

if (taskData[task] === 0) delete taskData[task];

}

saveTasks();

updateChart();

if (Object.keys(taskData).length === 0) {

document.getElementById("chartContainer").style.display = "none";

}

}

// Save tasks in localStorage

function saveTasks() {

localStorage.setItem("tasks", JSON.stringify(taskData));

localStorage.setItem("taskColors", JSON.stringify(colors));

}

// Load tasks from localStorage on page load

function loadTasks() {

let ul = document.getElementById("taskList");

ul.innerHTML = "";

for (let task in taskData) {

let li = document.createElement("li");

li.innerHTML = `${task} <span class='delete' onclick='removeTask(this, "${task}")'>✖</span>`;

ul.appendChild(li);

}

if (Object.keys(taskData).length > 0) {

document.getElementById("chartContainer").style.display = "block";

updateChart();

}

}

// Update chart based on stored data

function updateChart() {

let ctx = document.getElementById("taskChart").getContext("2d");

if (window.taskChartInstance) window.taskChartInstance.destroy();

window.taskChartInstance = new Chart(ctx, {

type: "pie",

data: {

labels: Object.keys(taskData),

datasets: [{

data: Object.values(taskData),

backgroundColor: Object.keys(taskData).map(task => colors[task]),

}],

},

options: {

plugins: {

legend: {

position: 'right',

labels: {

color: 'white',

}

}

}

}

});

}

**DASHBOARD:**

document.addEventListener("DOMContentLoaded", function() {

let workoutData = JSON.parse(localStorage.getItem("workoutData")) || {};

let dietData = JSON.parse(localStorage.getItem("dietData")) || {};

updateChart("workoutChart", workoutData);

updateChart("dietChart", dietData);

});

let charts = {}; // Store chart instances globally

function updateChart(id, storedData) {

let ctx = document.getElementById(id).getContext("2d");

let labels = Object.keys(storedData); // Dates

let data = labels.map(date => {

let values = storedData[date];

return Array.isArray(values) ? values.reduce((a, b) => a + b, 0) : values;

});

if (charts[id]) {

charts[id].destroy(); // Destroy previous chart instance before updating

}

charts[id] = new Chart(ctx, {

type: "bar",

data: {

labels: labels,

datasets: [{

label: id === "workoutChart" ? "Workout Duration (mins)" : "Water Intake (ml)",

data: data,

backgroundColor: "rgba(0, 255, 255, 0.5)"

}]

},

options: {

scales: { y: { beginAtZero: true } }

}

});

}

function addWorkout() {

let duration = document.getElementById("duration").value;

if (duration) {

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

let workoutData = JSON.parse(localStorage.getItem("workoutData")) || {};

if (!workoutData[today]) {

workoutData[today] = [];

}

workoutData[today].push(parseInt(duration));

localStorage.setItem("workoutData", JSON.stringify(workoutData));

updateChart("workoutChart", workoutData); // ✅ Update graph instantly

}

}

function updateDietChart(storedData) {

let ctx = document.getElementById("dietChart").getContext("2d");

let labels = Object.keys(storedData); // Get dates

let caloriesData = labels.map(date => storedData[date].calories);

let waterData = labels.map(date => storedData[date].water);

if (charts["dietChart"]) {

charts["dietChart"].destroy(); // Destroy previous chart instance

}

charts["dietChart"] = new Chart(ctx, {

type: "bar",

data: {

labels: labels, // ✅ Display Dates as labels

datasets: [

{

label: "Calories Intake (kcal)",

data: caloriesData,

backgroundColor: "rgba(255, 99, 132, 0.5)"

},

{

label: "Water Intake (ml)",

data: waterData,

backgroundColor: "rgba(0, 255, 255, 0.5)"

}

]

},

options: {

scales: {

x: { title: { display: true, text: "Date" } }, // ✅ Add x-axis title

y: { beginAtZero: true, title: { display: true, text: "Amount" } } // ✅ Add y-axis title

}

});

}

**STORAGE:**

// Load saved data when the page loads

window.onload = function () {

loadData();

};

// Store data in LocalStorage

function saveData(key, value) {

localStorage.setItem(key, JSON.stringify(value));

}

// Load data from LocalStorage

function loadData() {

if (localStorage.getItem("workoutData")) {

document.getElementById("exercise").value = JSON.parse(localStorage.getItem("workoutData")).exercise || "";

document.getElementById("duration").value = JSON.parse(localStorage.getItem("workoutData")).duration || "";

}

if (localStorage.getItem("dietData")) {

document.getElementById("caloriesMeal").value = JSON.parse(localStorage.getItem("dietData")).calories || "";

document.getElementById("water").value = JSON.parse(localStorage.getItem("dietData")).water || "";

}

if (localStorage.getItem("caloriesData")) {

let data = JSON.parse(localStorage.getItem("caloriesData"));

document.getElementById("age").value = data.age || "";

document.getElementById("height").value = data.height || "";

document.getElementById("weight").value = data.weight || "";

document.getElementById("caloriesResult").innerText = data.calories || "--";

}

}

// Track Workout and Store Data

function addWorkout() {

let exercise = document.getElementById("exercise").value;

let duration = document.getElementById("duration").value;

if (!exercise || !duration) {

alert("Please fill out both fields!");

return;

}

let workoutData = { exercise, duration };

saveData("workoutData", workoutData);

alert("Workout data saved!");

}

// Track Diet & Water Intake and Store Data

function addDiet() {

let calories = document.getElementById("caloriesMeal").value;

let water = document.getElementById("water").value;

if (!calories || !water) {

alert("Please fill out both fields!");

return;

}

let dietData = { calories, water };

saveData("dietData", dietData);

alert("Diet data saved!");

}

// Calculate and Store Calories

function calculateCalories() {

let age = document.getElementById("age").value;

let height = document.getElementById("height").value;

let weight = document.getElementById("weight").value;

if (!age || !height || !weight) {

alert("Please fill out all fields!");

return;

}

let calories = Math.round((10 \* weight) + (6.25 \* height) - (5 \* age) + 5);

document.getElementById("caloriesResult").innerText = calories;

let caloriesData = { age, height, weight, calories };

saveData("caloriesData", caloriesData);

alert("Calories data saved!");

}

// Simulate Heartbeat Tracking

function trackHeartbeat() {

let bpm = Math.floor(Math.random() \* (120 - 60) + 60);

document.getElementById("bpm").innerText = bpm;

}