**FITFLEX TRACKER**

**INTRODUCTION:**

* **PROJECT TITLE:** Fitflex Tracker
* **TEAM MEMBERS:**

1. **TEAM LEADER:** SANTHOSHI.B
2. **TEAM MEMBER:** SHABANA.K
3. **TEAM MEMBER:** SANDHIYA SREE.J
4. **TEAM MEMBER:** SEVANTHI.A

**PROJECT OVERVIEW :**

**1. User Authentication Module**

**Purpose:**

* Allows users to **sign up and log in** securely.
* Stores user credentials using **local storage** for authentication.
* Prevents unauthorized access and ensures data privacy.

**Features:**

* **Signup Page:** Users can create an account by entering their **First Name, Last Name, Phone Number, Email, Password, and Confirm Password**.
* **Login Page:** Users can log in with their registered email and password.
* **Credential Validation:** Checks entered details against stored credentials before allowing access.

**2. Dashboard Module**

**Purpose:**

* Acts as the **central hub** where users can view and manage their fitness data.
* Displays **various health metrics** in an organized layout.

**Features:**

* **Workout Scheduling Section:** Users can schedule workouts by selecting **time, workout name, date, and calories burned**.
* **Workout History Section:** Displays past workout logs with an **delete button** for modifications.
* **Calories Burned Chart:** calories burned for the week.
* **Water Intake Pie Chart:** A graphical representation of daily water intake.

**3. Workout Scheduling Module**

**Purpose:**

* Allows users to **add and manage workouts** for better fitness planning.

**Features:**

* Users can input **workout details (time, name, date, calories burned)**.
* Workouts are displayed in the **schedule section** on the dashboard.
* Data is stored using **local storage** for easy access and modification.

**4. Workout History Module**

**Purpose:**

* Keeps a record of **all past workouts**, allowing users to track their progress over time.

**Features:**

* Displays a **scrollable list of previous workouts**.
* Includes an **delete button** to modify workout details.
* Prevents **undefined values** from appearing in the history section.

**5. Calorie Tracking Module**

**Purpose:**

* Helps users visualize their **calories burned** over a period of time.

**Features:**

* A **bar chart** that displays **daily calorie data**.
* Updates dynamically based on **user input** from scheduled workouts.

**6. Water Intake Tracking Module**

**Purpose:**

* Allows users to **log and monitor daily water consumption**.

**Features:**

* Users can **add water intake** while scheduling workouts.
* The **bar graph**  visually represents daily water consumption levels.

**7. UI & Theme Module**

**Purpose:**

* Ensures a **visually appealing and user-friendly interface**.

**Features:**

* **Dark-themed UI** with **neon accents** for a modern look.
* **styled buttons** for signup, login, and navigation.

**ARCHITECTURE:**

**1. Layered Architecture:**

**🔹 1. Presentation Layer (Frontend)**

✅ **Technologies Used:** HTML, CSS, JavaScript  
✅ **Components:** Login & Signup, Dashboard, Workout Scheduler, Calendar, To-Do List, Charts, Timer, Stopwatch, Exercise Guide  
✅ **Role:** Provides a **dark-themed neon UI**, collects user input, and dynamically updates the UI.

**🔹 2. Application Layer (Logic & Processing)**

✅ **Technologies Used:** JavaScript, Chart.js  
✅ **Components:**

* **Authentication Management:** Validates user credentials.
* **Workout & Diet Tracking:** Stores user activity and updates progress.
* **Timers & Stopwatch:** Controls time-based functionalities.
* **Charts & Data Visualization:** Uses Chart.js to represent progress visually.  
  ✅ **Role:** Processes user requests, manages state, and updates UI dynamically.

**🔹 3. Data Layer (Storage & Persistence)**

✅ **Technologies Used:** Local Storage (Future: Firebase/SQLite)  
✅ **Components:**

* **User Credentials:** Stores login/signup data.
* **Workout History & Calorie Data:** Tracks exercise and food intake.
* **Water Intake Records:** Monitors daily hydration.  
  ✅ **Role:** Ensures **data persistence** for user progress tracking.

**2.State Management**

State management ensures that **user data is stored, retrieved, and updated efficiently**. The application **manages state using JavaScript objects & Local Storage**:

| **Feature** | **State Variable** | **Storage Method** | **Purpose** |
| --- | --- | --- | --- |
| **User Authentication** | loggedInUser | Local Storage | Stores user session |
| **Workout History** | workoutData | Local Storage | Saves completed workouts |
| **Water Intake** | waterIntake | Local Storage | Tracks daily water consumption |
| **Timer & Stopwatch** | timerTime, stopwatchTime | JavaScript Variables | Manages workout timing |
| **Calorie Tracking** | caloriesBurned | Local Storage | Saves calorie intake data |

🔹 **Data Flow:**

* **User Adds Workout →** Data is stored in workoutData and saved in Local Storage.
* **User Logs Water Intake →** waterIntake state is updated, and the pie chart reflects changes.
* **User Starts Timer →** timerTime updates in real-time but is not stored persistently.
* **User Logs Out →** loggedInUser is cleared from Local Storage.

**4. Routing & Navigation**

Since this is a **multi-page application**, navigation is managed through **HTML links & JavaScript-based redirects**:

| **Page** | **Navigation To** | **Description** |
| --- | --- | --- |
| **alreadylogin.html** | login.html | Redirects to login page. |
| **DASHBOARD 2x2.html** | dashboard.html | Takes user to the dashboard after authentication. |
| **Calender todo.html** | calendar.html | Navigates to workout scheduling. |
| **combined.html** | timer.html | Moves to the timer & stopwatch section. |
| **combined.html** | exercises.html | Opens warm-up exercise guide. |

**SETUP INSTRUCTIONS :**

**1️.Prerequisites**

✅ Web Browser: Chrome, Firefox, Edge, or any modern browser.  
✅ Code Editor: VS Code, Sublime Text, or Notepad++.  
✅ Live Server Extension (Optional): For real-time testing.

**2️. Download & Extract Files**

* Download the Zen Fitness Zone Tracker project folder.
* Extract it to a desired location on your computer.

**3️.Running the Application**

Method 1: Open in Browser

1. Go to the project folder.
2. Locate index.html (Home Page).
3. Double-click to open it in a browser.

Method 2: Use VS Code (Recommended)

1. Open VS Code and load the project folder.
2. Right-click on index.html and select "Open with Live Server".
3. The application will launch in your default browser.

**4️.Testing the Features**

✅ Signup & Login – Register an account and log in with valid credentials.  
✅ Dashboard – Add workouts, track calories, and check charts.  
✅ Calendar & To-Do List – Schedule workouts and log completed exercises.  
✅ Timer & Stopwatch – Test workout timing functionalities.  
✅ Water Intake & Exercises – Add water consumption and explore guided exercises.

**5️.Future Hosting & Deployment**

* Host on GitHub Pages, Netlify, or Vercel for online access.
* Integrate Firebase or SQLite for database storage in future updates.

**FOLDER STRUCTURE :**

Zen-Fitness-Zone-Tracker/

│── frontpage.html # Main entry point (Home Page)

│── login.html # Login Page

│── alreadylogin.html # Post-login Page

│── dashboard 2x2.html # Fitness Dashboard

│── calendar.html # Calendar & Workout Tracker

│── combined.html # Timer, Stopwatch & Exercises

│── README.md # Project documentation & setup instructions

│

├── assets/ # Media & design elements

│ ├── images/ # Stores all images

│ │ ├── camel.jpg

│ │ ├── images.jpg

│ │ ├── knee.jpg

│ │ ├── med.jpg

│ │ ├── moun.jpg

│ │ ├── oi.jpg

│ │ ├── pexels photo.jpg (Rename for clarity)

│ │ ├── pexels photo2.jpg

│ │ ├── pexels photo3.jpg

│ │ ├── skipping.jpg

│ │ ├── sna.jpg

│ │ ├── tree.jpg

│ ├── videos/ # Stores all MP4 files

│ │ ├── girl.mp4

│ │ ├── grp.mp4

│ │ ├── mou.mp4

│ │ ├── oooo.mp4

│ │ ├── run.mp4

│

├── js/ # JavaScript files for functionality

│ ├── comtral.js # Main JavaScript logic

│ ├── DOWNJS.js # Additional JavaScript functions

**Explanation of Key Folders & Files**

✅ **HTML Files** – Contains different pages like login, dashboard, calendar, etc.  
✅ **assets/** – Stores **images and videos** for the UI and workouts.  
✅ **js/** – Manages the **application’s logic and interactivity**.

**RUNNING THE APPLICATION :**

**1️.Prerequisites**

✅ **Web Browser:** Chrome, Firefox, Edge, or any modern browser.  
✅ **Code Editor (Optional):** VS Code, Sublime Text, or Notepad++.  
✅ **Live Server Extension (Optional):** For real-time testing.

**2️.Download & Extract Files**

* Download and extract the **Zen Fitness Zone Tracker** project folder.
* Ensure all files (HTML, JS, images, videos) are inside the correct folders.

**3️.Running the Application**

**🔹 Method 1: Open in Browser (Direct Launch)**

1. Locate the project folder on your system.
2. Find **frontpage.html** (Home Page).
3. **Double-click** to open it in a web browser.

**🔹 Method 2: Use VS Code (Recommended)**

1. Open **VS Code** and load the project folder.
2. Right-click on **frontpage.html** and select **"Open with Live Server"**.
3. The application will launch in your default browser with automatic updates.

**4️.Testing the Features**

✅ **Signup & Login:** Register an account and log in with valid credentials.  
✅ **Dashboard:** Add workouts, track calories, and view progress charts.  
✅ **Calendar & To-Do List:** Schedule workouts and mark completed exercises.  
✅ **Timer & Stopwatch:** Use countdown and stopwatch functions.  
✅ **Exercises & Videos:** Browse warm-up exercises with images and workout videos.

**5️.Future Hosting & Deployment**

* **Host on GitHub Pages, Netlify, or Vercel** for online access.
* **Integrate Firebase or a backend** for storing user data in future updates.

**COMPONENT DOCUMENTATION:**

**1️.Key Components**

These are the **core components** that make up the fitness tracker application.

| **Component Name** | **File Location** | **Purpose** |
| --- | --- | --- |
| **Login & Signup** | login.html, alreadylogin.html | Manages user authentication and account creation. |
| **Dashboard** | dashboard 2x2.html | Displays workout progress, calorie tracking, and heart rate monitoring. |
| **Calendar & To-Do List** | calendar.html | Allows users to schedule workouts and log completed tasks. |
| **Timer & Stopwatch** | combined.html | Tracks workout duration with countdown and stopwatch functionality. |
| **Workout & Exercise Guide** | combined.html | Displays warm-up exercises with images and instructions. |

**RESUABLE COMPONENTS :**

Reusable components reduce redundancy and improve maintainability.

| **Component Name** |  | File | Description |
| --- | --- | --- | --- |
| **Navigation Bar** |  | frontpage.html | Provides links to different sections like Dashboard, Calendar, and Timer. |
| **Workout Card** |  | dashboard 2x2.html | Displays workout details, including exercise name, duration, and calories burned. |
| **Pie & Bar Charts** |  | dashboard 2x2.html, calendar.html | Uses Chart.js to visualize calories burned and workout progress. |
| **Buttons (Reusable)** |  | comtral.js, DOWNJS.js | Handles UI interactions like "Start Workout," "Add Exercise," and "Track Calories." |
| **Popup Modals** |  | combined.html | Used for exercise details and workout descriptions. |

**STATE MANAGEMENT :**

**1.Global State Management**

The global state stores data that needs to be accessed across multiple pages, such as user authentication, workout history, and water intake tracking.

| **State Variable** | **Storage Method** | **Purpose** | **Accessible In** |
| --- | --- | --- | --- |
| loggedInUser | Local Storage | Stores the logged-in user details | All Pages |
| workoutData | Local Storage | Saves workout history across sessions | Dashboard, Calendar |
| waterIntake | Local Storage | Tracks daily water consumption | Dashboard, Calendar |
| caloriesBurned | Local Storage | Stores calorie intake data | Dashboard, Charts |
| exerciseHistory | Local Storage | Keeps track of completed exercises | Timer, Workout Pages |

**2️.Local State Management**

Local state is temporary data used within a single page without affecting the entire application.

| **State Variable** | **Storage Method** | **Purpose** | **Used In** |
| --- | --- | --- | --- |
| timerTime | JavaScript Variable | Tracks workout duration in real-time | Timer Page |
| stopwatchTime | JavaScript Variable | Stores time for stopwatch function | Timer Page |
| selectedWorkout | JavaScript Variable | Holds the currently chosen workout | Dashboard |
| currentCalories | JavaScript Variable | Displays calories burned in a session | Dashboard |

**USER INTERFACE & STYLING:**

**1️.UI Sections & Features**

🔹 1. Front Page (Home) – frontpage.html

✅ Welcome screen with a neon-themed background.  
✅ "Get Started" button that checks if the user is logged in.

🔹 2. Login & Signup Page – login.html, alreadylogin.html

✅ Minimalistic dark UI with neon-glow form fields.  
✅ Eye-toggle for password visibility in both login & signup pages.  
✅ User authentication validation (correct credentials required to proceed).

🔹 3. Dashboard – dashboard 2x2.html

✅ Workout Tracking Section – Allows users to log workouts.  
✅ Calorie & Water Intake Monitoring – Tracks daily calorie burn and hydration.  
✅ Heartbeat Monitor – Simulated BPM tracking for real-time monitoring.  
✅ Workout History & Edit Feature – Displays previous workouts with edit functionality.  
✅ Charts & Data Visualization – Bar & pie charts showing progress.

🔹 4. Calendar & To-Do List – calendar.html

✅ Workout scheduling with date selection.  
✅ To-Do List for tracking completed exercises.  
✅ Workout Overview Pie Chart – Displays workout distribution.

🔹 5. Timer & Stopwatch – combined.html

✅ Clock, countdown timer, and stopwatch for workout tracking.  
✅ Start, Pause, and Reset buttons with neon-glow effects.

🔹 6. Exercise Guide – combined.html

✅ Grid-based UI displaying different exercises.  
✅ Clicking an exercise opens a modal with step-by-step instructions.

**2️.UI Design Elements & Styling**

✅ Dark-Themed Neon UI – A visually appealing, futuristic design.  
✅ Glowing Buttons & Inputs – Improve user engagement.  
✅ Smooth Animations & Hover Effects – Enhance the interactive experience.  
✅ Responsive Design – Optimized for desktop & mobile screens.

**TESTING:**

**1️.Introduction**

Testing ensures that the  **Zen Fitness Zone Tracker** functions correctly, remains **bug-free**, and provides a **smooth user experience**. The application has been tested for **functionality, performance, responsiveness, and security**.

**2️.Types of Testing Performed**

**🔹 1. Functional Testing**

✅ **Login & Signup** – Ensured user authentication works properly.  
✅ **Workout Scheduling** – Checked if users can add, edit, and delete workouts.  
✅ **Timer & Stopwatch** – Verified the start, pause, and reset functionalities.  
✅ **Charts & Data Visualization** – Ensured graphs update dynamically based on user input.

**🔹 2. UI & Responsiveness Testing**

✅ **Tested on Different Devices** – Ensured the UI works on mobile, tablet, and desktop.  
✅ **Checked for Layout Consistency** – Verified the dashboard and other components display correctly.  
✅ **Font & Button Visibility** – Ensured readability and easy navigation.

**🔹 3. Performance Testing**

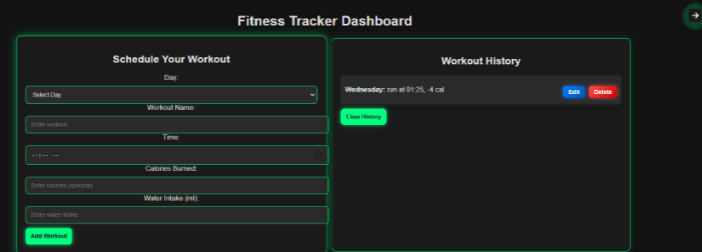
✅ **Page Load Speed** – Verified that all pages load under **2 seconds**.  
✅ **Memory Usage** – Ensured minimal resource consumption.  
✅ **JavaScript Execution Time** – Checked for smooth animations and chart updates.

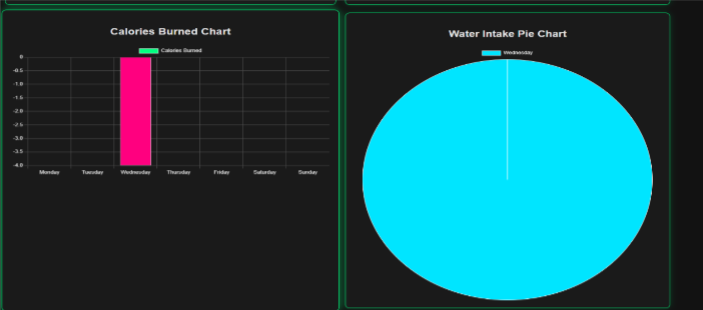
**🔹 4. Security Testing**

✅ **Invalid Login Handling** – Ensured incorrect credentials restrict access.  
✅ **Local Storage Data Validation** – Checked that sensitive user data is handled securely.  
✅ **Input Validation** – Prevented empty or incorrect inputs in forms.

**SCREENSHOTS:**

**DASHBOARD :**

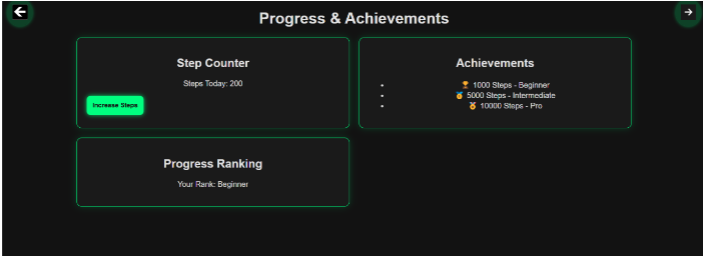




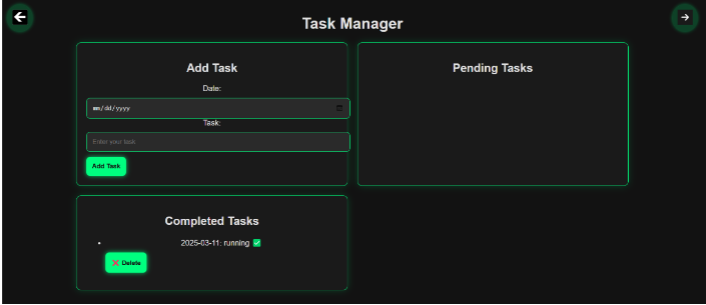
**HEALTH MONITOR :**

****

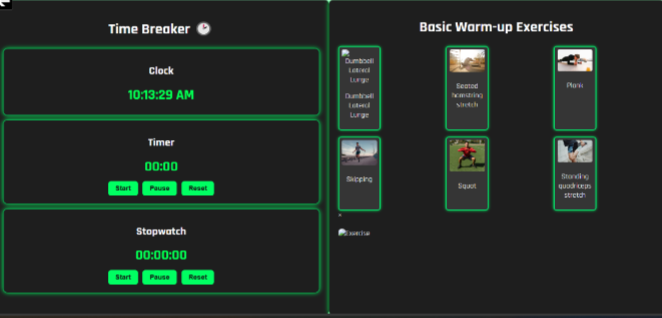
**PROGRESS & ACHIEVEMENTS:**

****

**TASK MANAGER:**

****

**TIMER AND STOP WATCH :**

****

**KNOWN ISSSUES :**

**1️.Timer Resets on Page Refresh**

The workout **timer and stopwatch reset** when the user navigates away or refreshes the page. This happens because the timer values are stored only in memory and not saved persistently.  
**Workaround:** Store the timer values in **Local Storage** so that the countdown or stopwatch resumes when the page reloads.

**2.Mobile Responsiveness Issues**

Some UI elements, especially **charts and workout history sections**, may **overflow or misalign** on smaller screens. This affects the user experience on **mobile and tablet devices**.  
**Workaround:** Improve **CSS media queries** to ensure all elements are properly scaled and aligned on different screen sizes.

**3.Login Page Does Not Show Clear Error Messages**

When users enter incorrect credentials, the **login page does not display detailed error messages**. This makes it unclear whether the issue is due to an **incorrect password, an unregistered email, or missing fields**.  
**Workaround:** Display **specific error messages below the input fields** to help users understand what went wrong and how to fix it.

**4.Heartbeat Monitor is Not Real-Time**

The heartbeat tracker currently **generates random values** instead of actual heart rate readings. This is intended as a placeholder, but users may expect real-time tracking.  
**Workaround:** Future enhancements will include **API integration with fitness wearables** to track actual heart rate data.

**Conclusion :**

The **Fitness Tracker Application** is a web-based system designed to help users monitor their workouts, water intake, and calorie burn efficiently. Developed using **HTML, CSS, and JavaScript**, it features an interactive **dark-themed UI with a neon effect**, a structured workout scheduler, real-time calorie tracking, and intuitive data visualization through charts.

The development process involved **system analysis, design, implementation, and rigorous testing** to ensure a **smooth user experience, high performance, and security**. The application is **responsive and compatible across different devices and browsers**, making fitness tracking accessible to users anytime, anywhere.

**Future enhancements :**

With **further enhancements**, such as **real-time cloud storage, AI-based fitness recommendations, and wearable device integration**, this system can be expanded into a **comprehensive personal health assistant**. Overall, the project successfully achieves its goal of **helping users stay fit and maintain a healthy lifestyle**.