



COMPLETE CODING TUTORIAL & SETUP GUIDE

# MediHealth Website Tutorial

Learn how to build a complete healthcare web application from scratch with HTML, CSS, JavaScript, and Supabase.

# Contents

- Part 1** — Project Overview
- Part 2** — Building the HTML
- Part 3** — Styling with CSS
- Part 4** — JavaScript Interactivity
- Part 5** — Supabase Setup
- Part 6** — Database & Connection
- Part 7** — Authentication
- Part 8** — GitHub
- Part 9** — Testing

---

This tutorial covers the key code for building and deploying MediHealth. It focuses on the important snippets, not every line.

# Part 1: Project Overview

Three files in the same folder:

File	Job	Think of it as...
<code>index.html</code>	Structure	The skeleton
<code>styles.css</code>	Design	The skin
<code>app.js</code>	Interactivity	The brain

`index.html` — linking the files

```
<!-- In the <head> section -->
<link rel="stylesheet" href="styles.css">
<script src="https://cdn.jsdelivr.net/npm/@supabase/supabase-js@2"></script>

<!-- Before </body> at the very bottom -->
<script src="app.js"></script>
```

# Part 2: Building the HTML

## 2.1 — Navigation

Stays at the top. Each link uses #sectionId to jump to that section.

index.html — Navigation

```
<header class="header" id="header">
  <nav class="nav">

    <!-- Logo -->
    <a href="#" class="logo">
      <div class="logo__icon">+</div>
      <span class="logo__text">MediHealth</span>
    </a>

    <!-- Nav links -->
    <ul class="nav__menu">
      <li><a href="#hero">Home</a></li>
      <li><a href="#services">Services</a></li>
      <li><a href="#booking">Book</a></li>
      <li><a href="#tips">Health Tips</a></li>
      <li><a href="#account">Account</a></li>
    </ul>

    <!-- Hamburger (mobile only) -->
    <button class="nav__burger" id="navBurger">
      <span></span>
      <span></span>
      <span></span>
    </button>

  </nav>
</header>
```

The three <span> elements become horizontal bars via CSS. JavaScript rotates them into an X on click.

## 2.2 — Booking Form

Every input has an id so JavaScript can grab its value.

index.html — Booking form

```
<form id="bookingForm">

  <!-- Two inputs side-by-side -->
  <div class="form-row">
    <div class="form-group">
      <label>First Name *</label>
      <input type="text" id="bkFname" required>
    </div>
    <div class="form-group">
      <label>Last Name *</label>
      <input type="text" id="bkLname" required>
    </div>
  </div>

  <!-- Service dropdown -->
  <div class="form-group">
    <label>Service *</label>
    <select id="bkService" required>
      <option value="">Select a service...</option>
      <option value="general-checkup">General Check-up</option>
      <option value="blood-test">Blood Test</option>
    </select>
  </div>

  <button type="submit">Confirm Appointment</button>

  <!-- JS shows this with a success/error message -->
  <div class="form-msg" id="bkMsg"></div>

</form>
```

## 2.3 — Health Tips (Interactive)

Chips have **data-cat** attributes. Grid starts empty — JS fills it.

index.html — Health tips

```
<div class="tips__filters" id="tipFilters">
  <button class="chip active" data-cat="all">All Tips</button>
  <button class="chip" data-cat="nutrition">Nutrition</button>
  <button class="chip" data-cat="fitness">Fitness</button>
  <button class="chip" data-cat="sleep">Sleep</button>
</div>

<!-- Empty — JavaScript fills this with cards -->
<div class="tips__grid" id="tipsGrid"></div>
```

## 2.4 — Testimonial Carousel

Write cards once. JS duplicates them for infinite scrolling.

index.html — Carousel

```
<div class="carousel">
  <div class="carousel__track" id="carouselTrack">

    <div class="testimonial">
      <div class="testimonial__stars">★★★★★</div>
      <p class="testimonial__quote">
        "MediHealth made booking so easy..."
      </p>
      <div class="testimonial__author">
        <div class="testimonial__avatar">ER</div>
        <div>
          <div class="testimonial__name">Emily Richardson</div>
          <div class="testimonial__role">Patient since 2024</div>
        </div>
      </div>
    </div>

    <!-- Repeat for more testimonials -->

  </div>
</div>
```

## 2.5 — Account Section

index.html — Account

```
<!-- Logged out -->
<div id="authPanel">
  <button data-tab="login">Sign In</button>
  <button data-tab="signup">Create Account</button>
  <form id="loginForm">...</form>
  <form id="signupForm">...</form>
</div>

<!-- Logged in (hidden by default) -->
<div id="userPanel" style="display:none">
  <div id="uName">User Name</div>
  <div id="uAppointments">...</div>
  <button onclick="MediHealth.signOut()">Sign Out</button>
</div>
```

## Part 3: Styling with CSS

### 3.1 — CSS Variables

styles.css — Variables

```
:root {
  --teal:      #2A9D8F; /* Primary colour */
  --teal-dark: #1B7A6E; /* Button colour */
  --cream:     #FAFAF5; /* Background */
  --ink:       #1A1F16; /* Main text */
}

body {
  background: var(--cream);
  color: var(--ink);
}
```

Change one variable and the whole site updates.

### 3.2 — Fixed Nav with Frosted Glass

styles.css — Navigation

```
.header {
  position: fixed; /* Stays at top */
  top: 0; left: 0; right: 0;
  z-index: 9000; /* Above everything */
  backdrop-filter: blur(24px); /* Frosted glass */
  background: rgba(250,250,245, 0.82);
}

/* JS adds this class on scroll */
.header.scrolled {
  padding: 12px 48px;
  box-shadow: 0 2px 8px rgba(0,0,0,.06);
}
```

### 3.3 — Grid Layouts + Responsive

styles.css — Responsive grid

```
/* Desktop: 3 columns */
.services__grid {
  display: grid;
  grid-template-columns: repeat(3, 1fr);
  gap: 24px;
}

/* Tablet: 1 column */
@media (max-width: 1080px) {
  .services__grid {
    grid-template-columns: 1fr;
  }
}
```

## 3.4 — Carousel Animation

styles.css — Carousel

```
.carousel {
  overflow: hidden;
  mask-image: linear-gradient(
    90deg, transparent, black 8%, black 92%, transparent
  );
}

.carousel__track {
  display: flex;
  gap: 24px;
  animation: scroll 35s linear infinite;
}

.carousel__track:hover {
  animation-play-state: paused;
}

@keyframes scroll {
  0% { transform: translateX(0); }
  100% { transform: translateX(-50%); }
}
```

## 3.5 — Scroll Reveal

styles.css — Scroll reveal

```
.scroll-reveal {
  opacity: 0;
  transform: translateY(28px);
  transition: all 0.55s ease;
}

/* JS adds .visible → fades in */
.scroll-reveal.visible {
  opacity: 1;
  transform: translateY(0);
}
```



# Part 4: JavaScript

## 4.1 — Module Pattern

app.js — Module structure

```
const MediHealth = (() => {

  let supabaseClient = null;
  let currentUser = null;

  function init() {
    initNavigation();
    initScrollEffects();
    initCarousel();
    renderTips("all");
    initTipFilters();
    initForms();
  }

  return { init, connectSupabase, signOut };

})();

document.addEventListener("DOMContentLoaded", MediHealth.init);
```

## 4.2 — Scroll Reveal

app.js — Scroll animations

```
function initScrollEffects() {
  const observer = new IntersectionObserver((entries) => {
    entries.forEach(entry => {
      if (entry.isIntersecting) {
        entry.target.classList.add("visible");
      }
    });
  }, { threshold: 0.12 });

  document.querySelectorAll(".scroll-reveal")
    .forEach(el => observer.observe(el));
}
```

## 4.3 — Health Tips Filtering

app.js — Tips filtering

```
const healthTips = [
  { category: "nutrition", title: "Balanced Eating", body: "... " },
  { category: "fitness", title: "30-Min Movement", body: "... " },
];

function renderTips(category) {
  const filtered = category === "all"
    ? healthTips
    : healthTips.filter(t => t.category === category);

  document.getElementById("tipsGrid").innerHTML =
    filtered.map(tip => `
      <div class="tip-card">
        <h4>${tip.title}</h4>
        <p>${tip.body}</p>
      </div>
    `).join("");
}
```

## 4.4 — Carousel Duplication

The simplest function. Copies all cards so the CSS loop is seamless.

app.js — Infinite carousel

```
function initCarousel() {
  const track = document.getElementById("carouselTrack");
  track.innerHTML = track.innerHTML + track.innerHTML;
}
```

## 4.5 — Booking Submission

app.js — Booking

```
async function handleBooking(e) {
  e.preventDefault();

  const data = {
    first_name: document.getElementById("bkFname").value,
    last_name: document.getElementById("bkLname").value,
    email: document.getElementById("bkEmail").value,
    service: document.getElementById("bkService").value,
    appointment_date: document.getElementById("bkDate").value,
    appointment_time: document.getElementById("bkTime").value,
    status: "confirmed"
  };

  const { error } = await supabaseClient
    .from("appointments")
    .insert([data]);

  if (error) showMessage(msg, "error", error.message);
  else showMessage(msg, "success", "Confirmed!");
}
```

## 4.6 — Authentication

app.js — Auth

```
// Sign up
await supabaseClient.auth.signUp({
  email, password,
  options: { data: { first_name, last_name } }
});

// Sign in
await supabaseClient.auth.signInWithPassword({
  email, password
});

// Sign out
await supabaseClient.auth.signOut();
```

## Part 5: Setting Up Supabase

### 1 Create Account

Go to [supabase.com](https://supabase.com) and sign up.

### 2 Create Project

Click **New Project**, name it, choose a password and region.

### 3 Get Credentials

Go to **Settings > API**. Copy **Project URL** and **anon key**.

## Part 6: Database & Connection

### 1 Create Table

In **SQL Editor**, paste and run the CREATE TABLE SQL.

### 2 Add Security

Run the Row Level Security policies SQL.

### 3 Connect

In `index.html`, paste URL and key into the hidden inputs.

## Part 7: Authentication

### 1 Turn Off Email Confirmation

**Authentication > Sign In / Providers** → toggle **Confirm email OFF**.

**Tip:** Supabase built-in email often fails. Turn it off for instant sign-up.

### 2 Clean Up

Delete unconfirmed users in **Authentication > Users**.

## Part 8: GitHub

1

## Create Repo

[github.com/new](https://github.com/new) → name it, set Public, README on.

2

## Upload

Add file > Upload → drag in all 4 files → Commit.

## Part 9: Testing & Troubleshooting

1.	Navigation	Links scroll. Active highlights. Mobile menu.
2.	Booking	Submits. Green message. Data in Supabase.
3.	Reschedule/Cancel	Lookup works. Buttons update DB.
4.	Health Tips	Display. Chips filter correctly.
5.	Carousel	Scrolls. Pauses on hover. Loops.
6.	Auth	Sign up/in works. Shows appointments.
7.	Responsive	Correct at all screen sizes.

### Common Problems

Problem	Fix
Demo mode	Check hidden inputs have Supabase credentials
Email error on signup	Turn off Confirm email in Supabase
CSS not loading	styles.css must be same folder as index.html
Blank page	Press F12, check Console for errors
Lookup empty	Re-run security policies SQL

**You now have a fully working MediHealth website. Well done!**