

Partea 1: „Creierul ascuns” și Setările (Secțiunea <head>)

Aici se dau instrucțiunile pentru browser. Nimic de aici nu apare direct pe ecran, în afară de titlul din tab-ul de sus.

```
<!DOCTYPE html>
<html lang="ro">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Dashboard - Constantin Lupu</title>
<link href="https://fonts.googleapis.com/css2?family=Lato..."
rel="stylesheet">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.4.0/css/all.min.css">
```

- <!DOCTYPE html>: Îi spui browserului să folosească cel mai nou standard de HTML (HTML5).
- <html lang="ro">: Setezi limba întregului document. Ajută Google-ul să știe că site-ul e în română.
- <meta charset="UTF-8">: Activează suportul pentru diacritice (ă, î, ș, ț, â) și caractere speciale.
- <meta name="viewport" ...>: **Esențial pentru telefoane.** Forțează site-ul să își adapteze lățimea la ecranul dispozitivului, nu să se micșoreze ca o imagine statică.
- <title>: Titlul care apare în bara de sus a browserului.
- <link>-urile: Primul importă fonturile frumoase (Lato și Playfair) de la Google Fonts. Al doilea aduce *Font Awesome* – o librărie care îți permite să pui iconițe (cum sunt cele de Facebook, TikTok) folosind doar o clasă de text, fără să descarci imagini pentru ele.

Partea 2: Designul și Hainele Site-ului (Secțiunea <style>)

Aici definim CSS-ul, adică modul în care arată totul.

1. Setările generale și Fundalul

CSS

```
* { box-sizing: border-box; }  
body {  
  background-image: url('Imagini/Deshboard/avif\ 1.avif');  
  background-attachment: fixed;  
  display: flex; flex-direction: column; min-height: 100vh;  
}
```

- **box-sizing: border-box:** O regulă de aur. Face ca marginile interioare (padding) să nu modifice lățimea totală a unui element, prevenind stricarea sablonului.
- **background-attachment: fixed:** Creează acel efect superb unde dai scroll la site, dar imaginea din spate rămâne nemișcată (efect de paralaxă).
- **display: flex . . .:** Face ca tot corpul site-ului (body) să funcționeze ca o coloană elastică. Dacă site-ul nu are mult text, **min-height: 100vh** obligă pagina să fie măcar cât înălțimea ecranului, ținând subsolul (footer) mereu jos.

CSS

```
.profile-link { position: absolute; top: -75px; left: 50%; transform: translateX(-50%); z-index: 10; }  
.profile-avatar { border-radius: 50%; border: 5px solid white; transition: transform 0.3s; }  
.profile-avatar:hover { transform: scale(1.05); }
```

- Aici ai făcut o șmecherie vizuală. În mod normal, poza ar sta înăuntrul headerului. Folosind **position: absolute** și **top: -75px**, ai ridicat-o deasupra marginii. **left: 50%** împreună cu **transform: translateX(-50%)** o centrează matematic, perfect pe mijloc.

border-radius: 50% face poza perfect rotundă.

- **:hover:** Când pui mouse-ul pe ea, **transform: scale(1.05)** o mărește puțin și lin (datorită la **transition: 0.3s**).

3. Meniul (Bara de navigare)

CSS

```
.navbar { position: sticky; top: 0; z-index: 100; }  
.nav-links a:hover { transform: scale(1.05); }
```

- `position: sticky; top: 0;` Când dai scroll în jos, meniul se va deplasa odată cu pagina până ajunge în tavan (`top: 0`), moment în care se „lipește” acolo ca să îl ai mereu la dispoziție. `z-index: 100` îl pune deasupra tuturor celorlalte elemente ca să nu fie acoperit.

4. Adaptabilitatea pentru Telefon (Responsivitatea)

CSS

```
@media (max-width: 900px) { ... }
```

- Orice cod CSS pus între aceste acolade se va activa **DOAR** dacă lățimea ecranului scade sub 900 de pixeli (telefoane, tablete mici). Aici muți elementele din header unele sub altele (`flex-direction: column`), micșorezi poza de profil și ascunzi meniul orizontal pentru a-l înlocui cu butonul „Hamburger”.

Partea 3: Scheletul Vizibil (Secțiunea `<body>` în HTML)

Aici pui efectiv cutiile pe pagină, cutii pe care le-ai decorat mai sus în CSS.

1. Header-ul

HTML

```
<header>  
  <a href="index.html" class="profile-link">...</a>  
  
  <div class="header-content-wrapper">  
    <div class="header-info">  
      <h1>Constantin Lupu</h1>  
      <a href="..." class="social-icon"><i  
class="fab fa-facebook-f"></i></a>  
    </div>  
  
    <div class="header-video-container">
```

```

        <video autoplay loop muted playsinline
controls>...</video>
    </div>

    <div class="header-time-container">
        <div id="clock">00:00:00</div>
        <div id="date">...</div>
    </div>
</div>
</header>

```

- Această zonă folosește Flexbox (header-content-wrapper). Elementele din ea (Info, Video, Ceas) sunt puse unele lângă altele automat, cu spațiu între ele (gap: 40px din CSS).
- Pentru <video>, atributele autoplay (pornește singur), muted (fără sunet) și loop (se repetă) sunt obligatorii; fără muted, browserele moderne ar bloca redarea automată.
- Observă id="clock" și id="date". Acelea sunt ca niște „găleți goale” pe care le va umple JavaScript-ul mai târziu.

2. Meniul

HTML

```

<nav class="navbar">
    <div class="menu-toggle" id="mobile-
menu">...</div>
    <div class="nav-links" id="nav-links">
        <a href="index.html" class="active">Acasă</a>
        ...
    </div>
</nav>

```

- menu-toggle este format din 3 span-uri (liniute). Acesta va fi butonul tău de pe mobil.
- Link-urile te duc spre celelalte pagini. Clasa active îi arată utilizatorului pe ce pagină se află în acel moment (e colorată diferit).

3. Șablonul Principal (Layout) și Sidebar-ul

HTML

```

<div class="layout-wrapper">
    <aside class="sidebar">
        <strong id="vizitatori">128</strong>
    </aside>
    ...
</div>

```

```
</aside>
<main style="flex: 1;"></main>
</div>
```

- Partea principală este despărțită în două. `<aside>` (o bară laterală, stânga) unde ai pus statisticile și widget-ul extern de la hitwebcounter.
 - `<main>` este momentan gol. Acolo urmează să îți pui conținutul paginii „Acasă”. Datorită `flex: 1`; în CSS, el va ocupa tot spațiul rămas liber din dreapta sidebar-ului.
-

Partea 4: Motoarele site-ului (Secțiunea `<script>`)

JavaScript-ul dă viață paginii. Fără el, ceasul ar fi doar un text static.

1. Funcționalitatea meniului pe mobil

```
JavaScript
const mobileMenuBtn =
document.getElementById('mobile-menu');
const navLinks = document.getElementById('nav-
links');

if(mobileMenuBtn) {
  mobileMenuBtn.addEventListener('click', () => {
    navLinks.classList.toggle('active');
  });
}
```

- Găsește butonul de mobil și meniul ascuns. Când dai *click* pe buton, adaugă/șterge clasa `active` pe meniu. În CSS, meniul normal e ascuns, dar meniul care primește clasa `active` apare pe ecran!

2. Ceasul și Data

```
JavaScript
function updateClock() {
  const now = new Date(); // Ia ora calculatorului
  // Extrage ore, minute, secunde si adauga zero-
  uri (ex: 09 in loc de 9)
  let h = now.getHours().toString().padStart(2,
'0');
  // ...
```

```

    document.getElementById('clock').textContent =
`${h}:${m}:${s}`;
}

```

- Aici iei ora exactă din sistem, o formatezi frumos și o bagi direct în div-ul cu id="clock" pe care îl aveam mai sus în HTML. Mai jos în cod există setInterval, care rulează funcția asta o dată la 1000 milisecunde (o secundă), astfel ceasul „ticăie”.

3. Timpul Cumulat (LocalStorage)

```

JavaScript
const STORAGE_KEY = 'site_constantin_total_time';
let totalSeconds = 0;

if (localStorage.getItem(STORAGE_KEY)) {
    totalSeconds =
parseInt(localStorage.getItem(STORAGE_KEY)); //
Incarca timpul anterior
} else {
    totalSeconds = 49374; // Daca esti utilizator
nou, porneste de la numarul asta
}

```

- localStorage este o bază de date micuță încorporată în browser. Când cineva intră pe site, codul verifică dacă a mai fost. Dacă da, reia timpul de unde a rămas. Dacă nu, îi dă o valoare fixă de start. La fel ca ceasul, acest timp crește cu +1 la fiecare secundă.

4. Iluzia de Trafic (Vizitatori și Mesaje)

```

JavaScript
function simulateVisitors() {
    if (Math.random() > 0.7) {
        vizitatoriCurenti++;
        const el =
document.getElementById('vizitatori');
        el.textContent = vizitatoriCurenti;
        el.classList.remove('fade-update');
        void el.offsetWidth; // TRUC VIZUAL
        el.classList.add('fade-update');
    }
}
setInterval(simulateVisitors, 3000);

```

- La fiecare 3 secunde, se rulează un „zar” (`Math.random`). Dacă numărul pică peste 0.7 (deci ai 30% șanse), numărul vizitatorilor crește.
- Aici intervine un truc vizual mișto: pentru ca animația de „sclipire” (`fade-update`) să pornească de fiecare dată, trebuie să ștergi clasa, să forțezi browserul să înțeleagă că ai șters-o (asta face acea linie ciudată `void el.offsetWidth;`) și să pui clasa la loc.

Acesta este tot ecosistemul! Un șablon extrem de solid, logic împărțit.