

Despre mine

1. Fundația paginii (Declarații și Meta Tag-uri)

HTML

```
<!DOCTYPE html>
<html lang="ro">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-
scale=1.0">
<title>Despre Mine - Constantin Lupu</title>
```

- **<!DOCTYPE html>**: Îi spune browserului (Chrome, Safari, etc.) că folosești cea mai nouă versiune de HTML (HTML5).
- **<html lang="ro">**: Indică faptul că tot conținutul site-ului este în limba română. Ajută motoarele de căutare (Google) și programele care citesc ecranul pentru persoanele nevăzătoare.
- **<meta charset="UTF-8">**: Permite afișarea corectă a caracterelor speciale și a diacriticelor românești (ă, î, ș, ț, â).
- **<meta name="viewport"...>**: **Foarte important!** Fără linia asta, site-ul tău ar arăta pe telefon exact ca pe PC, doar că foarte mic. Această linie îi spune browserului să adapteze lățimea paginii la lățimea ecranului dispozitivului.
- **<title>**: Textul care apare sus de tot, pe tab-ul (fila) browserului tău.

2. Resurse externe (Fonturi și Iconițe)

HTML

```
<link
href="https://fonts.googleapis.com/css2?family=Lato:wght@300;
400;700&family=Playfair+Display:wght@700&display=swap"
rel="stylesheet">
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.4.0/css/all.min.css">
```

- **Prima linie (<link href="..."...Google Fonts>):** Aici importi fonturi de pe internet. Am ales "Lato" (pentru textul normal) și "Playfair Display" (pentru titluri). Parametrii precum wght@300;400;700 aduc grosimi diferite ale textului (subțire, normal, îngroșat).
- **A doua linie (FontAwesome):** Este o bibliotecă uriașă de iconițe. Această linie îmi permite ca pe viitor să pun o iconiță de Facebook sau Instagram adăugând doar un cod scurt de genul <i class="fab fa-facebook"></i>.

3. Resetarea globală și Fundalul (Începe magia CSS)

CSS

```
<style>
```

```
/* --- RESET SI STILURI DE BAZA --- */
* { box-sizing: border-box; }
```

```
body {
  margin: 0;
  padding: 0;
  font-family: 'Lato', sans-serif;
  background-image: url('Imagini/Deshboard/avif\ 1.avif');
  background-size: cover;
  background-position: center;
  background-attachment: fixed;
  /* ... */
  display: flex;
  flex-direction: column;
  min-height: 100vh;
}
```

- *** { box-sizing: border-box; }:** Regula de aur în web design. Face ca dimensiunile pe care le dai unui element (de exemplu lățime 100px) să includă și marginile interioare (padding). Fără ea, elementele tale s-ar mări neașteptat.
- **margin: 0; padding: 0; pe body:** Browserele adaugă automat o margine albă în jurul paginii. Această comandă o

șterge, lăsând site-ul să se lipească perfect de marginile ecranului.

- **background-image... cover... fixed:** Aici am pus poza de fundal. **cover** îi spune pozei să acopere tot ecranul (chiar dacă trebuie să o taie un pic). **fixed** creează acel efect spectaculos prin care fundalul stă pe loc când dai scroll la text.
- **min-height: 100vh;** **vh** înseamnă "viewport height" (înălțimea ecranului). Garantează că site-ul tău va avea mereu o înălțime de minim 100% din ecranul vizibil, astfel încât secțiunea de subsol (footer) să nu se ridice la mijlocul paginii dacă ai puțin text.

4. Magia Pozei de Profil "Ieșite în afară"

CSS

```
.profile-avatar {  
  width: 150px;  
  height: 150px;  
  border-radius: 50%;  
  object-fit: cover;  
  border: 5px solid white;  
  position: absolute;  
  top: -75px;  
  left: 50%;  
  transform: translateX(-50%);  
  box-shadow: 0 4px 10px rgba(0,0,0,0.2);  
}
```

- **border-radius: 50%;** Transformă o poză pătrată într-un cerc perfect.
- **object-fit: cover;** Dacă poza ta originală nu e perfect pătrată, această regulă previne turtirea ei. O decupează frumos ca să umple cercul.
- **position: absolute; top: -75px;** Permite pozei să "zboare" peste restul elementelor. Fiind setată la -75px (jumătate din înălțimea ei de 150px), jumătate din ea iese în afara cutiei în care se află.

- **left: 50%; transform: translateX(-50%);**: Un truc clasic de CSS pentru a centra absolut un element pe orizontală.

5. Bara de Navigare (Meniul lipicios)

CSS

```
.navbar {  
    background-color: rgba(144, 164, 174, 0.9);  
    position: sticky;  
    top: 0;  
    z-index: 100;  
    /* ... */  
}  
.nav-links a:hover {  
    color: #fff;  
    background: rgba(255,255,255,0.2);  
    transform: scale(1.05);  
}
```

- **position: sticky; top: 0;**: Face ca bara de navigare să funcționeze normal până ajungi cu scroll-ul la ea, moment în care se "lipește" de tavanul ecranului (top: 0).
- **z-index: 100;**: Se asigură că meniul tău stă deasupra tuturor celorlalte elemente de pe pagină când dai scroll (ca straturile de foi dintr-un dosar).
- **:hover**: Reprezintă starea "când pui cursorul mouse-ului pe el". Când faci asta, link-ul își schimbă culoarea și se mărește ușor cu 5% (scale(1.05)).

6. Adaptabilitatea pe Telefoane Mobile (Media Queries)

CSS

```
@media (max-width: 900px) {  
    .main-container {  
        flex-direction: column;  
        /* ... */  
    }  
  
    .nav-links {
```

- ```

 display: none;
 flex-direction: column;
}

.menu-toggle {
 display: flex;
}
}

```
- **@media (max-width: 900px):** Este "întrerupătorul" pentru dispozitive mobile. Când ecranul scade sub 900 de pixeli lăţime, se activează acest cod care suprascrie codul de dinainte.
  - **flex-direction: column;** Pe PC, zona ta cu text şi zona cu video stăteau stânga-dreapta. Pe telefon, această comandă le pune una sub alta (în coloană).
  - **display: none; pe .nav-links:** Ascunde meniul clasic pe care îl vezi pe ecran lat, pentru că pe telefon nu ar avea loc.
  - **display: flex; pe .menu-toggle:** Afişează cele 3 liniuţe (hamburgerul) care pe PC erau ascunse.

---

## 7. HTML-ul Semnificativ (Corpul vizibil)

### HTML

```

<main class="main-container">
 <div class="text-column">
 </div>

 <div class="image-column">
 <video class="feature-image" autoplay muted loop
playsinline controls>
 <source src="video/mp 4.mp4" type="video/mp4">
 </video>
 </div>
</main>

```

- **<main>**: Un tag semantic care îi spune browserului (şi Google-ului): "Acesta este conţinutul principal al paginii".

- **<div>**: Cel mai folosit element HTML. Este pur și simplu o cutie (un container) pe care o folosești ca să grupezi alte elemente.
- **<video autoplay muted loop playsinline controls>**:
  - **autoplay**: pornește video-ul automat.
  - **muted**: fără sunet (este obligatoriu ca video-ul să fie pe "mut" pentru ca browserele moderne să îi dea voie să pornească automat).
  - **loop**: se repetă la nesfârșit.
  - **playsinline**: foarte util pentru iPhone-uri, previne video-ul din a se deschide automat pe tot ecranul când dai play.

---

## 8. Logica din JavaScript (Cum funcționează butonul de meniu)

HTML

<script>

```
const mobileMenuBtn = document.getElementById('mobile-menu');
```

```
const navLinks = document.getElementById('nav-links');
```

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

}

</script>

- **const ... = document.getElementById(...)**: Aici JavaScript-ul caută în HTML elementele de care are nevoie pe baza ID-urilor lor (id="mobile-menu" și id="nav-links"). E ca și cum i-ai spune: "Memorează acest buton ca să îl putem folosi".
- **.addEventListener('click', ...)**: JavaScript-ul se pune pe așteptare și "ascultă" când utilizatorul dă click pe acel buton.
- **.classList.toggle('active')**: Acesta este motorul acțiunii. Când dai click, adaugă clasa "active" pe meniu (ceea ce, în

CSS, are setat display: flex și face meniul să apară). Dacă dai click din nou, toggle va scoate clasa, ascunzând meniul.