

## Prezentare Moara Vlasiei

Acesta este un cod foarte interesant și mai avansat! Am creat ceea ce în web design se numește un layout de tip **"Web App" (Aplicație Web)**. Diferența principală față de pagina precedentă este că aici am blocat scroll-ul pe toată pagina (overflow: hidden) și am creat zone specifice care se derulează independent (textul și lista de poze). Este un design foarte modern pentru o galerie foto. Hai să "disecăm" codul pe secțiuni, ca să înțelegem exact mecanica din spate:

---

### 1. Structura de tip "Aplicație" (CSS-ul de pe body)

CSS

```
body {  
    /* ... */  
    height: 100vh;  
    display: flex;  
    flex-direction: column;  
    overflow: hidden;  
}
```

- **De ce este special?** Prin height: 100vh; forțezi pagina să ocupe exact 100% din înălțimea ecranului, niciun pixel mai mult.
- **overflow: hidden;** Aceasta este comanda care "taie" orice iese în afara ecranului și interzice scroll-ul general pe pagină.
- **display: flex; flex-direction: column;** Împarte ecranul pe verticală în 3 "felii": Header-ul sus, Galeria în mijloc (care ia tot spațiul rămas) și Footer-ul jos.

---

### 2. Scena principală (Zona unde apare poza mare)

CSS

```
.display-area {  
    flex: 1;  
    background-size: cover;  
    background-position: center;  
    transition: background-image 0.5s ease-in-out;
```

```

    /* ... */
}
.overlay {
    background: linear-gradient(to bottom, rgba(0,0,0,0) 30%,
    rgba(0,0,0,0.85) 100%);
    pointer-events: none;
}

```

- **.display-area:** flex: 1 îi spune să se întindă și să ocupe tot spațiul disponibil din stânga listei de miniaturi. transition face ca trecerea de la o poză la alta să fie lină (durează o jumătate de secundă), nu bruscă.
- **.overlay:** Este un filtru (o peliculă) pus peste poză. linear-gradient creează o trecere de la complet transparent (sus) la negru 85% opac (jos). Aceasta se face pentru ca textul alb să poată fi citit perfect, indiferent cât de luminoasă este poza din spate.
- **pointer-events: none;** Îi spune mouse-ului să "ignore" acest overlay, permițând utilizatorului să selecteze textul de dedesubt.

---

### 3. Detaliile estetice (Scrollbar personalizat și Animații)

#### CSS

```

.text-content::-webkit-scrollbar { width: 5px; }
.text-content::-webkit-scrollbar-thumb { background:
    rgba(255,255,255,0.4); ... }

@keyframes fadeIn { from { opacity: 0; } to { opacity: 1; } }
@keyframes slideIn { from { transform: translateX(-20px);
    opacity: 0; } to { ... } }

```

- **::-webkit-scrollbar:** Aici suprascrii designul urât, standard, al barei de scroll de la Windows/Mac. Am creat o bară subțire, elegantă și semitransparentă, special pentru zona de text și lista de poze.
- **@keyframes:** Aici declari animațiile (cum să apară textul când dai click pe o poză nouă). fadeIn face textul să apară din neant, iar slideIn îl face să vină ușor dinspre stânga.

---

## 4. HTML-ul interactiv (Declanșarea acțiunilor)

HTML

```
<div class="thumbnail-card" onclick="schimbaPoza('Imagini/...',  
'Palatul Ghica', 'Descriere...', this)">  
    
  <div class="thumb-title">Imaginea 2</div>  
</div>
```

- **onclick="..."**: Aceasta este legătura directă între HTML și JavaScript. Îi spui browserului: "Când cineva dă click pe această cutie, rulează funcția schimbaPoza".
- Tu trimiți 4 informații ("parametri") către JavaScript la fiecare click: 1. Calea către poză, 2. Titlul, 3. Descrierea, 4. this (care înseamnă "acest element HTML pe care tocmai s-a dat click", pentru a-i putea pune chenarul colorat).

---

## 5. Magia de pe Telefon (Media Query la max-width: 800px)

CSS

```
@media (max-width: 800px) {  
  .showcase-container { flex-direction: column; }  
  .thumbnail-list {  
    flex-direction: row;  
    overflow-x: auto;  
    /* ... */  
  }  
}
```

- Pe desktop, ai poza mare în stânga și lista de miniaturi în dreapta (flex-direction: row implicit).
- Pe telefon, prin flex-direction: column, muți panoul lateral tocmai jos.
- **.thumbnail-list**: Modifici lista de miniaturi dintr-una verticală (overflow-y: auto de pe desktop) într-un carusel orizontal (flex-direction: row și overflow-x: auto), pe care poți face "swipe" cu degetul stânga-dreapta.

---

## 6. Creierul Galeriei (JavaScript-ul)

## JavaScript

```
function schimbaPoza(imagine, titlu, descriere, element) {  
    // 1. Schimbă imaginea și textele  
    var displayArea = document.getElementById('main-bg');  
    displayArea.style.backgroundImage = "url('" + imagine +  
    "')";  
    document.getElementById('main-title').innerText = titlu;  
    document.getElementById('main-desc').innerText =  
    descriere;  
  
    // 2. Truc pentru a reporni animația  
    var textContent = document.querySelector('.text-content');  
    var clona = textContent.cloneNode(true);  
    textContent.parentNode.replaceChild(clona, textContent);  
  
    // 3. Logica de meniu activ  
    var thumbnails = document.querySelectorAll('.thumbnail-  
card');  
    thumbnails.forEach(function(card) {  
card.classList.remove('active'); });  
    if(element) element.classList.add('active');  
  
    // 4. Centrarea pe mobil  
    if (window.innerWidth <= 800) {  
        element.scrollIntoView({ behavior: 'smooth', block:  
'nearest', inline: 'center' });  
    }  
}
```

Aici se întâmplă logica grea:

1. Preia informațiile din HTML (poza, titlul, textul) și le "injectează" în scena principală.
2. **Truc avansat:** Când schimbi textul, animația CSS nu se declanșează a doua oară de la sine. Codul tău face o copie (clonă) a cutiei de text, o șterge pe cea veche și pune clona în loc. Asta păcălește browserul și îl forțează să ruleze animațiile (fadeIn și slideIn) de la capăt!

3. Caută toate miniaturile, le șterge clasa active (chenarul albastru) și o pune doar pe cea pe care tocmai ai dat click.
4. `scrollIntoView`: Dacă ești pe telefon și dai click pe o poză din marginea ecranului, acest cod mișcă automat caruselul orizontal pentru a aduce poza respectivă pe centrul ecranului. O funcție excelentă pentru experiența utilizatorului (UX).

---

Acest cod este practic o aplicație complet funcțională! Modul în care am combinat CSS Flexbox cu manipularea JavaScript a DOM-ului (Document Object Model) arată din punctul meu de vedere foarte profesionist/profesională.