Snippet: mockup telefono con demo live

Codice riutilizzabile per incorporare un'app navigabile dentro una cornice smartphone · Laboratorio EB Media Studio

A cosa serve

Incorporare un'applicazione web reale e navigabile dentro la cornice grafica di uno smartphone, in una pagina WordPress, senza che il CSS del tema ne rompa il layout e senza tagli sui bordi a nessuna risoluzione.

Come funziona

L'app vive su una sua pagina dedicata, isolata e in noindex. Il mockup la incorpora in un iframe dimensionato a misura di telefono (390x800) e poi rimpicciolito con `transform:scale`. La scala non è fissa: viene calcolata a runtime dalla larghezza reale del contenitore, così combacia sempre.

1. HTML

```
<div class="ph">
  <div class="ph-top"><span></span></div>
  <div class="ph-live">
    <iframe class="ph-frame" src="/demo/" title="Anteprima app"
            loading="lazy" scrolling="yes"></iframe>
  </div>
  <div class="ph-cap">Demo reale - scorri dentro lo schermo</div>
</div>
```

2. CSS

```
.ph{
  width:250px; max-width:250px;
  background:#0a1f1a; border:8px solid #0d2a22; border-radius:34px;
  margin:0 auto; display:flex; flex-direction:column; overflow:hidden;
  box-shadow:0 24px 60px rgba(0,0,0,.45);
}
.ph-top{height:26px;background:#0d2a22;display:flex;
  align-items:center;justify-content:center}
.ph-top span{width:56px;height:5px;border-radius:5px;
  background:rgba(255,255,255,.18)}

/* il rapporto d'aspetto fa seguire l'altezza alla larghezza:
   niente altezze scritte a mano, niente tagli in basso */
.ph-live{position:relative;overflow:hidden;width:100%;
  aspect-ratio:390/800;background:#f8f7f4}

/* !important neutralizza la regola globale iframe{max-width:100%}
   di molti temi: senza, l'iframe viene compresso PRIMA dello
   scaling e il layout interno dell'app si deforma */
.ph-frame{
  position:absolute; top:0; left:0;
  width:390px !important; height:800px !important;
  max-width:none !important;
  border:0; transform-origin:top left;
}
.ph-cap{background:#0d2a22;color:rgba(255,255,255,.5);font-size:10px;
  letter-spacing:.06em;text-align:center;padding:7px 6px;flex-shrink:0}

@media(max-width:860px){ .ph{width:230px;max-width:230px} }
```

3. JS: calcolo della scala a runtime

```
<script>
(function(){
  function fitPhoneFrames(){
```

```
    document.querySelectorAll('.ph-live').forEach(function(live){
      var frame = live.querySelector('.ph-frame');
      if(!frame) return;
      var scale = live.clientWidth / 390;   // 390 = larghezza base app
      frame.style.transform = 'scale(' + scale + ')';
    });
  }
  window.addEventListener('load', fitPhoneFrames);
  window.addEventListener('resize', fitPhoneFrames);
  setTimeout(fitPhoneFrames, 200);
  setTimeout(fitPhoneFrames, 800);
})();
<\/script>
```

4. Pagina della demo: isolamento totale dal tema

Se l'app è un singolo file HTML autonomo, incorporarlo con `srcdoc`: il contenuto vive in un documento separato e il CSS del tema non lo tocca.

```
<style>html,body{margin:0;padding:0;overflow:hidden}
  #f{position:fixed;inset:0;width:100%;height:100%;border:0}</style>
<iframe id="f" title="Demo"></iframe>
<script>
  /* B64 = l'intero file HTML dell'app, codificato in base64 */
  document.getElementById('f').srcdoc =
    decodeURIComponent(escape(atob(B64)));
<\/script>
```

Sulla pagina della demo aggiungere anche il meta robots `noindex,nofollow` e la disattivazione della barra di amministrazione tramite il filtro `show_admin_bar`.

Errori da non rifare

- Scala fissa per breakpoint (es. .64 desktop / .58 mobile): va fuori sincrono col contenitore e taglia bordo destro e inferiore.
- Dimenticare `max-width:none` sull'iframe: l'app viene compressa a metà larghezza e sbaglia i propri breakpoint.
- Fidarsi del ridimensionamento della finestra desktop per testare il mobile: il viewport reale può non cambiare. Verificare a 390px veri.
- Non svuotare la cache dell'hosting dopo la modifica.