

```

<!doctype html>

<html lang="en">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width,initial-scale=1" />
  <title>IBM-FE - Dynamic Image Slider (Single File)</title>
  <style>
    /* -----
      Reset & base
      -----*/
    *{box-sizing:border-box;margin:0;padding:0}
    :root{
      --bg:#0f1724; --card:#0b1220; --muted:#98a0b3; --accent:#06b6d4;
      --width:min(1100px, 94%);
    }
    body{font-family:Inter, ui-sans-serif, system-ui, -apple-system, "Segoe UI", Roboto, "Helvetica
  Neue", Arial; background:linear-gradient(180deg,var(--bg),#071025 120%); color:#e6eef7;
  min-height:100vh; display:flex; align-items:center; justify-content:center; padding:32px}/*
  -----
    Slider card
    -----*/
    .slider-wrap{width:var(--width); max-width:100%; background:linear-gradient(180deg,
  rgba(255,255,255,0.02), rgba(0,0,0,0.06)); border-radius:14px; padding:18px; box-shadow:0
  10px 30px rgba(2,6,23,0.6);}
    .header{display:flex;align-items:center;gap:12px;margin-bottom:12px}
    .logo{width:44px;height:44px;background:linear-gradient(135deg,var(--accent),#7c3aed);border-
  radius:10px;display:grid;place-items:center;font-weight:700}
    .title{font-size:16px}
    .subtitle{font-size:12px;color:var(--muted)}

    /* -----
      Viewport & slides
      -----*/
    .viewport{position:relative;overflow:hidden;border-radius:12px}
    .slides{display:flex;transition:transform 520ms cubic-bezier(.2,.9,.3,1);touch-action:pan-y}
    .slide{min-width:100%;position:relative}
    .slide img{width:100%;height:52vh;max-height:560px;object-fit:cover;display:block}

    /* Caption */
    .caption{position:absolute;left:18px;bottom:18px;background:linear-gradient(90deg,
  rgba(2,6,23,0.6), rgba(2,6,23,0.25));backdrop-filter:blur(6px);padding:10px
  14px;border-radius:10px}
    .caption h3{font-size:16px;margin-bottom:4px}
  
```

```

.caption p{font-size:13px;color:var(--muted)}

/* Prev/Next Buttons */
.btn{position:absolute;top:50%;transform:translateY(-50%);z-index:10;border:0;background:rgba(2,6,23,0.45);backdrop-filter:blur(6px);width:44px;height:44px;border-radius:10px;display:grid;place-items:center;cursor:pointer}
.btn:hover{background:rgba(2,6,23,0.6)}
.prev{left:12px}
.next{right:12px}
.btn svg{width:18px;height:18px;opacity:0.95}

/* Dots */
.dots{display:flex;gap:8px;position:absolute;left:50%;transform:translateX(-50%);bottom:12px;padding:6px 8px;background:rgba(2,6,23,0.22);border-radius:999px}
.dot{width:10px;height:10px;border-radius:50%;background:rgba(255,255,255,0.15);cursor:pointer}
.dot.active{background:var(--accent);box-shadow:0 0 8px rgba(6,182,212,0.18)}

/* Thumbnails row */
.thumbs{display:flex;gap:8px;margin-top:12px;overflow:auto;padding-bottom:6px}
.thumb{flex:0 0 110px;height:66px;border-radius:8px;overflow:hidden;cursor:pointer;position:relative;border:2px solid transparent}
.thumb img{width:100%;height:100%;object-fit:cover;display:block}
.thumb.active{border-color:var(--accent);box-shadow:0 6px 18px rgba(2,6,23,0.6)}

/* Controls row */
.controls{display:flex;align-items:center;justify-content:space-between;margin-top:10px}
.left-controls{display:flex;gap:10px;align-items:center}
.small-btn{background:#071b2a;padding:8px 12px;border-radius:8px;border:1px solid rgba(255,255,255,0.03);cursor:pointer}
.small-btn input{vertical-align:middle}

/* Responsive tweaks */
@media (max-width:640px){ .caption{left:12px;right:12px} .thumb{flex:0 0 84px} .slide img{height:40vh} }

</style>
</head>
<body>
  <main class="slider-wrap" aria-label="Dynamic image slider">
    <div class="header">
      <div class="logo">IBM</div>
      <div>

```

```

    <div class="title">IBM-FE Dynamic Image Slider</div>
    <div class="subtitle">Autoplay • Hover to pause • Keyboard & Touch support</div>
  </div>
  <div class="viewport" id="viewport">
    <button class="btn prev" id="prev" aria-label="Previous slide">
      <svg viewBox="0 0 24 24" fill="none" stroke="currentColor" stroke-width="2"><path d="M15
18l-6-6 6-6" stroke-linecap="round" stroke-linejoin="round"/></svg>
    </button>

    <div class="slides" id="slides">
      <!-- Slides: change image src or data-caption as needed -->
      <div class="slide" data-caption="Aurora over mountains">
        
        <div class="caption"><h3>Aurora Peaks</h3><p>Night sky and northern lights above the
range.</p></div>
      </div>

      <div class="slide" data-caption="Coastal sunrise">
        
        <div class="caption"><h3>Coastal Sunrise</h3><p>Golden hour at the cliffs.</p></div>
      </div>

      <div class="slide" data-caption="City skyline at dusk">
        
        <div class="caption"><h3>City Dusk</h3><p>Lights turning on across the
skyline.</p></div>
      </div>

      <div class="slide" data-caption="Forest path in autumn">
        
        <div class="caption"><h3>Autumn Path</h3><p>A quiet trail covered in leaves.</p></div>
      </div>

      <button class="btn next" id="next" aria-label="Next slide">
        <svg viewBox="0 0 24 24" fill="none" stroke="currentColor" stroke-width="2"><path d="M9
6l6 6 6 6" stroke-linecap="round" stroke-linejoin="round"/></svg>
      </button>

      <div class="dots" id="dots" aria-hidden="false"></div>
    </div>

```

```

<div class="thumbs" id="thumbs" aria-label="Slide thumbnails"></div>

<div class="controls">
  <div class="left-controls">
    <button class="small-btn" id="playPause" aria-pressed="false">Pause</button>
    <div style="font-size:13px;color:var(--muted)">Auto-play interval:
      <select id="intervalSelect"
style="margin-left:8px;background:transparent;border:0;color:inherit">
        <option value="2000">2s</option>
        <option value="3000" selected>3s</option>
        <option value="5000">5s</option>
      </select>
    </div>
  </div>
  <div style="font-size:13px;color:var(--muted)">Use ← → or swipe</div>
</div>

</main> <script>
/* -----
  Slider logic (Vanilla JS)
  Features:
  - Autoplay with pause on hover
  - Prev/Next buttons
  - Dots + thumbnails navigation
  - Keyboard and touch support
  - Lazy images (via loading="lazy")
  -----*/

(function(){
  const slidesEl = document.getElementById('slides');
  const slides = Array.from(slidesEl.children);
  const prevBtn = document.getElementById('prev');
  const nextBtn = document.getElementById('next');
  const dotsEl = document.getElementById('dots');
  const thumbsEl = document.getElementById('thumbs');
  const playPauseBtn = document.getElementById('playPause');
  const intervalSelect = document.getElementById('intervalSelect');
  const viewport = document.getElementById('viewport');

  let index = 0;
  let timer = null;
  let autoplay = true;
  let interval = parseInt(intervalSelect.value, 10) || 3000;

```

```

// Build dots & thumbs
slides.forEach((s, i) => {
  const d = document.createElement('button'); d.className='dot';
d.setAttribute('aria-label','Go to slide '+(i+1));
  d.addEventListener('click', ()=>goTo(i));
  dotsEl.appendChild(d);

  const t = document.createElement('div'); t.className='thumb'; t.tabIndex=0;
  const img = document.createElement('img'); img.src = s.querySelector('img').src; img.alt =
s.querySelector('img').alt || ('Slide '+(i+1));
  t.appendChild(img);
  t.addEventListener('click', ()=>goTo(i));
  t.addEventListener('keydown', (e)=>{ if(e.key==='Enter' || e.key===' ') { e.preventDefault();
goTo(i);} });
  thumbsEl.appendChild(t);
});

const dotEls = Array.from(dotsEl.children);
const thumbEls = Array.from(thumbsEl.children);

function refresh(){
  // move slides container
  slidesEl.style.transform = `translateX(${ -index * 100}%)`;
  dotEls.forEach((d, i)=> d.classList.toggle('active', i===index));
  thumbEls.forEach((t, i)=> t.classList.toggle('active', i===index));
}

function prev(){ index = (index - 1 + slides.length) % slides.length; refresh(); }
function next(){ index = (index + 1) % slides.length; refresh(); }
function goTo(i){ index = Math.max(0, Math.min(i, slides.length-1)); refresh(); resetTimer(); }

prevBtn.addEventListener('click', ()=>{ prev(); resetTimer(); });
nextBtn.addEventListener('click', ()=>{ next(); resetTimer(); });

// autoplay
function startTimer(){ if(timer) clearInterval(timer); timer = setInterval(()=>{ next(); }, interval);
}
function stopTimer(){ if(timer) { clearInterval(timer); timer = null; } }
function resetTimer(){ if(autoplay) { stopTimer(); startTimer(); } }

// Play/pause control
function updatePlayButton(){ playPauseBtn.textContent = autoplay ? 'Pause' : 'Play';
playPauseBtn.setAttribute('aria-pressed', String(!autoplay)); }

```

```
playPauseBtn.addEventListener('click', ()=>{ autoplay = !autoplay; if(autoplay) startTimer();  
else stopTimer(); updatePlayButton(); });
```

```
// interval change  
intervalSelect.addEventListener('change', (e)=>{ interval = parseInt(e.target.value,10);  
resetTimer(); });
```

```
// Pause on hover/focus
```

```
[viewport, prevBtn, nextBtn, playPauseBtn].forEach(el=>{  
  el.addEventListener('mouseenter', ()=>{ if(autoplay) stopTimer(); });  
  el.addEventListener('mouseleave', ()=>{ if(autoplay) startTimer(); });  
  el.addEventListener('focusin', ()=>{ if(autoplay) stopTimer(); });  
  el.addEventListener('focusout', ()=>{ if(autoplay) startTimer(); });  
});
```

```
// Keyboard navigation
```

```
window.addEventListener('keydown', (e)=>{  
  if(e.key==='ArrowLeft') { prev(); resetTimer(); }  
  if(e.key==='ArrowRight') { next(); resetTimer(); }  
});
```

```
// Touch / swipe support
```

```
let startX = 0, startTime = 0;  
slidesEl.addEventListener('pointerdown', (e)=>{ startX = e.clientX; startTime = Date.now();  
slidesEl.setPointerCapture(e.pointerId); });  
slidesEl.addEventListener('pointerup', (e)=>{  
  const dx = e.clientX - startX; const dt = Date.now() - startTime;  
  const threshold = Math.min(window.innerWidth * 0.12, 80);  
  if(Math.abs(dx) > threshold && dt < 600){ if(dx < 0) next(); else prev(); resetTimer(); }  
});
```

```
// Make first render
```

```
refresh();  
updatePlayButton();  
if(autoplay) startTimer();
```

```
// Accessibility: announce slide changes (simple)
```

```
const live = document.createElement('div'); live.setAttribute('aria-live','polite');  
live.style.position='absolute'; live.style.left='-9999px';  
document.body.appendChild(live);  
const observer = new MutationObserver(()=>{ live.textContent = `Slide ${index+1} of  
${slides.length}: ${slides[index].dataset.caption || ""}`; });  
observer.observe(slidesEl, {attributes:true, subtree:true, childList:true});  
})();
```

```
</script></body>  
</html>
```