SVG Texture Pack: Carbon Fibre, Chrome Vein, Snake Scale, Graphene Mesh

Below is a compact, production-ready texture pack you can drop into your app. It includes:

- Carbon Fibre Weave (twill-inspired, ultra-thin)
- Chrome Vein (micro, highlight-ready)
- Snake Scale (micro, metallic highlight)
- Graphene Hex Mesh (hex lattice)
- Ultra-fine Crosshatch Mesh (for micro-structure depth)

Each texture is tileable and designed to be layered at different opacities and gradient intensities across your UI.

Quick Start

Option A — Inline Sprite (best for SVG fills/masks):

1. Paste the "Sprite (Defs Only)" once in your HTML (ideally near the end of body).
2. Apply on any shape as `fill="url(#tex-carbon-fibre)"` etc.
3. Control intensity via the element's `opacity`, or pattern scaling via `patternTransform`.

Option B — Individual Tiles (best for CSS backgrounds):

1. Save the "Individual Tile" SVGs as files (e.g., `carbon-fibre.svg`).
2. Use in CSS as `background-image: url(carbon-fibre.svg); background-repeat: repeat;`
3. Layer multiple backgrounds and control per-layer opacity with alpha in color or overlay elements.

Sprite (Defs Only)

Embed once. Reference by `url(#PATTERN_ID)`.

```
<!-- textures-sprite.svg (embed once, e.g., in <body>) -->
<svg xmlns="http://www.w3.org/2000/svg" width="0" height="0" style="position:absolute;left:-
  <defs>
    <!-- Shared chrome highlight gradient -->
    <linearGradient id="grad-chrome" x1="0" y1="0" x2="128" y2="0" gradientUnits="userSpace(
      <stop offset="0" stop-color="#ffffff" stop-opacity="0.85"/>
      <stop offset="0.06" stop-color="#c9cdd1" stop-opacity="0.62"/>
      <stop offset="0.23" stop-color="#8a9196" stop-opacity="0.42"/>
```

```
  <stop offset="0.50" stop-color="#151719" stop-opacity="0.55"/>
  <stop offset="0.77" stop-color="#8a9196" stop-opacity="0.42"/>
  <stop offset="0.94" stop-color="#c9cdd1" stop-opacity="0.62"/>
  <stop offset="1" stop-color="#ffffff" stop-opacity="0.85"/>
</linearGradient>

<!-- Subtle noise (optional) -->
<filter id="noise-01" x="-20%" y="-20%" width="140%" height="140%">
  <feTurbulence type="fractalNoise" baseFrequency="0.9" numOctaves="2" seed="4" />
  <feColorMatrix type="saturate" values="0"/>
  <feComponentTransfer>
    <feFuncA type="linear" slope="0.02"/>
  </feComponentTransfer>
</filter>

<!-- 1) Carbon Fibre Weave (twill-like, micro) -->
<pattern id="tex-carbon-fibre" width="16" height="16" patternUnits="userSpaceOnUse">
  <rect width="16" height="16" fill="transparent"/>
  <g transform="rotate(45 8 8)" opacity="0.55">
    <rect x="-8" y="6.5" width="32" height="1.1" fill="#000"/>
    <rect x="-8" y="9.5" width="32" height="1.1" fill="#000" opacity="0.6"/>
    <rect x="-8" y="0.5" width="32" height="1.1" fill="#333" opacity="0.35"/>
    <rect x="-8" y="3.5" width="32" height="1.1" fill="#000" opacity="0.4"/>
  </g>
  <!-- optional noise for realism -->
  <rect x="0" y="0" width="16" height="16" fill="black" opacity="0" filter="url(#noise-0
</pattern>

<!-- 2) Chrome Vein (ultra-thin highlights) -->
<pattern id="tex-chrome-vein" width="128" height="128" patternUnits="userSpaceOnUse">
  <rect width="128" height="128" fill="transparent"/>
  <g fill="none" stroke-linecap="round">
    <path d="M-5,96 C18,84 36,76 62,78 C88,80 102,95 132,90" stroke="url(#grad-chrome)"
    <path d="M-10,40 C12,36 32,24 66,26 C98,28 110,50 138,48" stroke="url(#grad-chrome)'
    <path d="M-6,10 C22,22 54,16 78,18 C108,20 120,6 136,8" stroke="url(#grad-chrome)" s
    <path d="M-8,70 C8,62 26,56 54,58 C84,60 100,74 134,72" stroke="url(#grad-chrome)" s
  </g>
</pattern>

<!-- 3) Snake Scale (micro, metallic) -->
<radialGradient id="grad-scale" cx="50%" cy="30%" r="70%">
  <stop offset="0" stop-color="#ffffff" stop-opacity="0.35"/>
  <stop offset="0.35" stop-color="#9aa0a6" stop-opacity="0.22"/>
  <stop offset="0.7" stop-color="#2a2d30" stop-opacity="0.28"/>
  <stop offset="1" stop-color="#000000" stop-opacity="0.4"/>
</radialGradient>
```

```
    <pattern id="tex-snake-scale" width="48" height="28" patternUnits="userSpaceOnUse">
      <rect width="48" height="28" fill="transparent"/>
      <!-- scale 1 -->
      <path d="M24,2 C34,2 44,8 48,14 C44,20 34,26 24,26 C14,26 4,20 0,14 C4,8 14,2 24,2 Z"
      <path d="M0,14 C4,8 14,2 24,2 C34,2 44,8 48,14" fill="none" stroke="#ffffff" stroke-op
      <path d="M0,14 C4,20 14,26 24,26 C34,26 44,20 48,14" fill="none" stroke="#000000" stro
      <!-- scale 2 (shifted) -->
      <g transform="translate(24,14)">
        <path d="M24,2 C34,2 44,8 48,14 C44,20 34,26 24,26 C14,26 4,20 0,14 C4,8 14,2 24,2 Z
        <path d="M0,14 C4,8 14,2 24,2 C34,2 44,8 48,14" fill="none" stroke="#ffffff" stroke-
        <path d="M0,14 C4,20 14,26 24,26 C34,26 44,20 48,14" fill="none" stroke="#000000" st
      </g>
    </pattern>

    <!-- 4) Graphene Hex Mesh (tile: width=24, height 20.7846) -->
    <pattern id="tex-graphene" width="24" height="20.7846" patternUnits="userSpaceOnUse">
      <g fill="none" stroke="#c0ffd6" stroke-opacity="0.22" stroke-width="0.8" stroke-linejo
        <path d="M6,0 L18,0 L24,10.3923 L18,20.7846 L6,20.7846 L0,10.3923 Z"/>
        <path d="M18,0 L24,10.3923"/>
        <path d="M6,0 L0,10.3923"/>
      </g>
    </pattern>

    <!-- 5) Ultra-fine Crosshatch Mesh -->
    <pattern id="tex-mesh-ultra" width="8" height="8" patternUnits="userSpaceOnUse">
      <g opacity="0.25">
        <path d="M0,0 L8,8" stroke="#ffffff" stroke-opacity="0.06" stroke-width="0.6" stroke
        <path d="M8,0 L0,8" stroke="#ffffff" stroke-opacity="0.06" stroke-width="0.6" stroke
        <path d="M0,4 L8,4" stroke="#00ff00" stroke-opacity="0.06" stroke-width="0.5"/>
        <path d="M4,0 L4,8" stroke="#00ff00" stroke-opacity="0.06" stroke-width="0.5"/>
      </g>
    </pattern>
  </defs>
</svg>
```

How to use the sprite patterns on an overlay:

```
<!-- Example overlay using multiple layers -->
<svg viewBox="0 0 100 100" preserveAspectRatio="none" style="position:fixed;inset:0;pointer-
  <rect width="100%" height="100%" fill="url(#tex-carbon-fibre)" opacity="0.25"/>
  <rect width="100%" height="100%" fill="url(#tex-graphene)" opacity="0.15"/>
  <rect width="100%" height="100%" fill="url(#tex-chrome-vein)" opacity="0.18"/>
</svg>
```

Tip: Adjust density via `patternTransform="scale(0.8)"` on the pattern instance by cloning the `<pattern>` node with a new id and transform.

Individual Tiles (for CSS backgrounds)

Save the following as separate `.svg` files. They are single-tile images designed to repeat seamlessly with `background-repeat: repeat`. Use `background-size` to scale.

1) carbon-fibre.svg

```
<svg xmlns="http://www.w3.org/2000/svg" width="16" height="16" viewBox="0 0 16 16">
  <defs>
    <filter id="noise-01" x="-20%" y="-20%" width="140%" height="140%">
      <feTurbulence type="fractalNoise" baseFrequency="0.9" numOctaves="2" seed="4" />
      <feColorMatrix type="saturate" values="0"/>
      <feComponentTransfer><feFuncA type="linear" slope="0.02"/></feComponentTransfer>
    </filter>
  </defs>
  <rect width="16" height="16" fill="transparent"/>
  <g transform="rotate(45 8 8)" opacity="0.55">
    <rect x="-8" y="6.5" width="32" height="1.1" fill="#000"/>
    <rect x="-8" y="9.5" width="32" height="1.1" fill="#000" opacity="0.6"/>
    <rect x="-8" y="0.5" width="32" height="1.1" fill="#333" opacity="0.35"/>
    <rect x="-8" y="3.5" width="32" height="1.1" fill="#000" opacity="0.4"/>
  </g>
  <rect x="0" y="0" width="16" height="16" fill="black" opacity="0" filter="url(#noise-01)"/
</svg>
```

2) chrome-vein.svg

```
<svg xmlns="http://www.w3.org/2000/svg" width="128" height="128" viewBox="0 0 128 128">
  <defs>
    <linearGradient id="grad-chrome" x1="0" y1="0" x2="128" y2="0" gradientUnits="userSpaceO
      <stop offset="0" stop-color="#ffffff" stop-opacity="0.85"/>
      <stop offset="0.06" stop-color="#c9cdd1" stop-opacity="0.62"/>
      <stop offset="0.23" stop-color="#8a9196" stop-opacity="0.42"/>
      <stop offset="0.50" stop-color="#151719" stop-opacity="0.55"/>
      <stop offset="0.77" stop-color="#8a9196" stop-opacity="0.42"/>
      <stop offset="0.94" stop-color="#c9cdd1" stop-opacity="0.62"/>
      <stop offset="1" stop-color="#ffffff" stop-opacity="0.85"/>
    </linearGradient>
  </defs>
  <rect width="128" height="128" fill="transparent"/>
  <g fill="none" stroke-linecap="round">
    <path d="M-5,96 C18,84 36,76 62,78 C88,80 102,95 132,90" stroke="url(#grad-chrome)" stro
    <path d="M-10,40 C12,36 32,24 66,26 C98,28 110,50 138,48" stroke="url(#grad-chrome)" str
    <path d="M-6,10 C22,22 54,16 78,18 C108,20 120,6 136,8" stroke="url(#grad-chrome)" strok
    <path d="M-8,70 C8,62 26,56 54,58 C84,60 100,74 134,72" stroke="url(#grad-chrome)" strok
```

```
  </g>
</svg>
```

3) snake-scale.svg

```
<svg xmlns="http://www.w3.org/2000/svg" width="48" height="28" viewBox="0 0 48 28">
  <defs>
    <radialGradient id="grad-scale" cx="50%" cy="30%" r="70%">
      <stop offset="0" stop-color="#ffffff" stop-opacity="0.35"/>
      <stop offset="0.35" stop-color="#9aa0a6" stop-opacity="0.22"/>
      <stop offset="0.7" stop-color="#2a2d30" stop-opacity="0.28"/>
      <stop offset="1" stop-color="#000000" stop-opacity="0.4"/>
    </radialGradient>
  </defs>
  <rect width="48" height="28" fill="transparent"/>
  <path d="M24,2 C34,2 44,8 48,14 C44,20 34,26 24,26 C14,26 4,20 0,14 C4,8 14,2 24,2 Z" fill
  <path d="M0,14 C4,8 14,2 24,2 C34,2 44,8 48,14" fill="none" stroke="#ffffff" stroke-opacit
  <path d="M0,14 C4,20 14,26 24,26 C34,26 44,20 48,14" fill="none" stroke="#000000" stroke-o
  <g transform="translate(24,14)">
    <path d="M24,2 C34,2 44,8 48,14 C44,20 34,26 24,26 C14,26 4,20 0,14 C4,8 14,2 24,2 Z" fi
    <path d="M0,14 C4,8 14,2 24,2 C34,2 44,8 48,14" fill="none" stroke="#ffffff" stroke-opac
    <path d="M0,14 C4,20 14,26 24,26 C34,26 44,20 48,14" fill="none" stroke="#000000" stroke
  </g>
</svg>
```

4) graphene-mesh.svg

```
<svg xmlns="http://www.w3.org/2000/svg" width="24" height="20.7846" viewBox="0 0 24 20.7846"
  <rect width="24" height="20.7846" fill="transparent"/>
  <g fill="none" stroke="#c0ffd6" stroke-opacity="0.22" stroke-width="0.8" stroke-linejoin="
    <path d="M6,0 L18,0 L24,10.3923 L18,20.7846 L6,20.7846 L0,10.3923 Z"/>
    <path d="M18,0 L24,10.3923"/>
    <path d="M6,0 L0,10.3923"/>
  </g>
</svg>
```

5) mesh-ultra.svg

```
<svg xmlns="http://www.w3.org/2000/svg" width="8" height="8" viewBox="0 0 8 8">
  <rect width="8" height="8" fill="transparent"/>
  <g opacity="0.25">
    <path d="M0,0 L8,8" stroke="#ffffff" stroke-opacity="0.06" stroke-width="0.6"/>
    <path d="M8,0 L0,8" stroke="#ffffff" stroke-opacity="0.06" stroke-width="0.6"/>
    <path d="M0,4 L8,4" stroke="#00ff00" stroke-opacity="0.06" stroke-width="0.5"/>
    <path d="M4,0 L4,8" stroke="#00ff00" stroke-opacity="0.06" stroke-width="0.5"/>
  </g>
</svg>
```

CSS Usage Examples

Layer multiple textures with separate opacities and sizes:

```
:root {
  --tex-carbon: url('/img/textures/carbon-fibre.svg');
  --tex-graphene: url('/img/textures/graphene-mesh.svg');
  --tex-chrome: url('/img/textures/chrome-vein.svg');
  --tex-mesh: url('/img/textures/mesh-ultra.svg');
  --tex-snake: url('/img/textures/snake-scale.svg');
}

.app-surface {
  /* Layer order: topmost first */
  background-image:
    var(--tex-chrome),
    var(--tex-graphene),
    var(--tex-carbon);
  background-repeat: repeat, repeat, repeat;
  /* Control density per layer */
  background-size:
    220px 220px, /* chrome vein: looser */
    48px 41.57px,/* graphene: native tile size ~sqrt(3)*r*2 */
    16px 16px;   /* carbon fibre: tight */
  /* Optional blend modes if supported: */
  background-blend-mode: screen, overlay, multiply;
  /* Base */
  background-color: #000;
  color: #0f0; /* if you want green-on-black aesthetic */
}
```

Opacity control in CSS backgrounds:

- For backgrounds, opacity comes from the SVG's strokes/fills. To "dim" a layer, prefer a semi-transparent overlay:

```
.app-surface::after {
  content: "";
  position: absolute; inset: 0;
  background: var(--tex-mesh);
  background-size: 8px 8px;
  opacity: 0.15;       /* global opacity for the layer */
  mix-blend-mode: overlay;
  pointer-events: none;
}
```

Inline SVG Fill Examples

```
<!-- Card background with carbon fibre and graphene mixed -->
<svg viewBox="0 0 400 200" width="400" height="200" role="img" aria-label="Carbon fibre and
  <rect width="100%" height="100%" fill="#000"/>
  <rect width="100%" height="100%" fill="url(#tex-carbon-fibre)" opacity="0.28"/>
  <rect width="100%" height="100%" fill="url(#tex-graphene)" opacity="0.16"/>
  <rect width="100%" height="100%" fill="url(#tex-chrome-vein)" opacity="0.18"/>
</svg>
```

Scale any texture without editing paths by cloning the pattern and adding `patternTransform`:

```
<!-- Denser graphene (75% original scale) -->
<svg width="0" height="0">
  <defs>
    <pattern id="tex-graphene-dense" href="#tex-graphene" patternTransform="scale(0.75)"/>
  </defs>
</svg>
```

Recommended Layering Recipes

- Subtle high-tech base:
  - Carbon Fibre at opacity 0.18–0.28
  - Graphene Mesh at opacity 0.10–0.18
- Premium metallic highlight:
  - Chrome Vein at opacity 0.12–0.22 (screen or lighten blend)
- Organic/tech hybrid:
  - Snake Scale at opacity 0.12–0.20 beneath Chrome Vein 0.14

Tips

- Color theming:
  - Change the stroke colors inside the SVG (e.g., graphene's `stroke="#c0ffd6"`) to your app's accent.
  - For a green-on-black theme, keep black background and use neon-green strokes sparingly (opacity <= 0.2).
- Performance:
  - Prefer the individual tiles as CSS backgrounds for large surfaces.
  - Keep background-size close to tile size to avoid resampling overhead.
- Accessibility:

- Ensure sufficient contrast for content layers above textured backgrounds; add a solid or blurred underlay behind text.

If you'd like, tell me your app's primary color(s) and preferred density, and I can pre-tint and scale the pack specifically to your theme.