<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>AI Image Generator App</title>

</head>

<style>

    @import url('https://fonts.googleapis.com/css2?family=Poppins:wght@500&display=swap');

\* {

    margin: 0;

    padding: 0;

    font-family: 'Poppins', sans-serif;

}

body {

    background: linear-gradient(to bottom, #ffffff 0%, #e2e2e2 40%, #dcdcdc 150%), linear-gradient(to top, rgba(255,255,255,0.40) 0%, rgba(255, 255, 255, 0.25) 200%);

    background-blend-mode: multiply;

    background-repeat: no-repeat;

    background-attachment: fixed;

    background-size: cover;

}

.main {

    display: flex;

    justify-content: center;

    align-items: center;

    height: 100vh;

}

.container {

    display: flex;

    flex-direction: column;

    text-align: center;

    height: 100vh;

    width: 100vw;

}

.header > h1 {

    font-size: 60px;

}

.header > p {

    font-size: 22px;

}

.searchBox {

    display: flex;

    width: 100%;

    align-items: center;

    justify-content: space-between;

    gap: 8px;

    background: #eaeaea;

    border-radius: 50px;

    position: relative;

    margin-top: 20px;

}

.searchButton {

    color: white;

    position: absolute;

    right: 15px;

    width: 50px;

    height: 50px;

    border-radius: 50%;

    background: var(--gradient-2, linear-gradient(90deg, #333333 0%, #5e5e5e 100%));

    border: 0;

    display: inline-block;

    transition: all 300ms cubic-bezier(.23, 1, 0.32, 1);

    cursor: pointer;

}

button:hover {

    color: #fff;

    background-color: #1A1A1A;

    box-shadow: rgba(0, 0, 0, 0.5) 0 10px 20px;

    transform: translateY(-3px);

}

button:active {

    box-shadow: none;

    transform: translateY(0);

}

.searchInput {

    border: none;

    background: none;

    outline: none;

    color: rgb(0, 0, 0);

    font-size: 17px;

    width: 600px;

    padding: 24px 46px 24px 26px;

}

.image-gallery {

    display: flex;

    gap: 15px;

    padding: 0 15px;

    flex-wrap: wrap;

    justify-content: center;

    align-items: center;

    margin: 50px auto;

    height: 100%;

}

.image-gallery .img-card {

    display: flex;

    position: relative;

    align-items: center;

    justify-content: center;

    background: #f2f2f2;

    border-radius: 4px;

    overflow: hidden;

    aspect-ratio: 1 / 1;

    width: 320px;

}

.image-gallery .img-card img {

    height: 100%;

    width: 100%;

    object-fit: cover;

}

.image-gallery .img-card.loading img {

    width: 80px;

    height: 80px;

}

.image-gallery .img-card .download-btn {

    bottom: 15px;

    right: 15px;

    height: 36px;

    width: 36px;

    display: flex;

    align-items: center;

    justify-content: center;

    text-decoration: none;

    background: #fff;

    border-radius: 50%;

    position: absolute;

    opacity: 0;

    pointer-events: none;

    transition: 0.2s ease;

}

.image-gallery .img-card .download-btn img {

    width: 14px;

    height: 14px;

}

.image-gallery .img-card:not(.loading):hover .download-btn {

    opacity: 1;

    pointer-events: auto;

}

@media screen and (max-width: 760px) {

    .image-generator {

        height: 45vh;

        padding-top: 30px;

        align-items: flex-start;

    }

    .image-generator h1 {

        font-size: 1.8rem;

    }

    .image-generator p {

        font-size: 1rem;

    }

    .image-generator .generate-form {

        margin-top: 30px;

        height: 52px;

        display: block;

    }

    .generate-form .controls {

        height: 40px;

        margin-top: 15px;

        justify-content: end;

        align-items: center;

    }

    .generate-form .generate-btn[disabled] {

        opacity: 1;

    }

    .generate-form .img-quantity {

        color: #fff;

    }

    .generate-form .img-quantity option {

        color: #000;

    }

    .image-gallery {

        margin-top: 20px;

    }

    .image-gallery .img-card:not(.loading) .download-btn {

        opacity: 1;

        pointer-events: auto;

    }

}

@media screen and (max-width: 500px) {

    .image-gallery .img-card {

        width: 100%;

    }

}

</style>

<body>

    <div class="main">

        <div class="container">

            <div class="header">

                <div class="searchBox">

                    <input class="searchInput" type="text" name="" placeholder="Describe your image...">

                    <button class="searchButton" onclick="generateImage()">

                        <svg xmlns="http://www.w3.org/2000/svg" width="29" height="29" viewBox="0 0 29 29" fill="none">

                            <g clip-path="url(#clip0\_2\_17)">

                                <g filter="url(#filter0\_d\_2\_17)">

                                    <path d="M23.7953 23.9182L19.0585 19.1814M19.0585 19.1814C19.8188 18.4211 20.4219 17.5185 20.8333 16.5251C21.2448 15.5318 21.4566 14.4671 21.4566 13.3919C21.4566 12.3167 21.2448 11.252 20.8333 10.2587C20.4219 9.2653 19.8188 8.36271 19.0585 7.60242C18.2982 6.84214 17.3956 6.23905 16.4022 5.82759C15.4089 5.41612 14.3442 5.20435 13.269 5.20435C12.1938 5.20435 11.1291 5.41612 10.1358 5.82759C9.1424 6.23905 8.23981 6.84214 7.47953 7.60242C5.94407 9.13789 5.08145 11.2204 5.08145 13.3919C5.08145 15.5634 5.94407 17.6459 7.47953 19.1814C9.01499 20.7168 11.0975 21.5794 13.269 21.5794C15.4405 21.5794 17.523 20.7168 19.0585 19.1814Z" stroke="white" stroke-width="3" stroke-linecap="round" stroke-linejoin="round" shape-rendering="crispEdges"></path>

                                </g>

                            </g>

                            <defs>

                                <filter id="filter0\_d\_2\_17" x="-0.418549" y="3.70435" width="29.7139" height="29.7139" filterUnits="userSpaceOnUse" color-interpolation-filters="sRGB">

                                    <feFlood flood-opacity="0" result="BackgroundImageFix"></feFlood>

                                    <feColorMatrix in="SourceAlpha" type="matrix" values="0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 127 0" result="hardAlpha"></feColorMatrix>

                                    <feOffset dy="4"></feOffset>

                                    <feGaussianBlur stdDeviation="2"></feGaussianBlur>

                                    <feComposite in2="hardAlpha" operator="out"></feComposite>

                                    <feColorMatrix type="matrix" values="0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0.25 0"></feColorMatrix>

                                    <feBlend mode="normal" in2="BackgroundImageFix" result="effect1\_dropShadow\_2\_17"></feBlend>

                                    <feBlend mode="normal" in="SourceGraphic" in2="effect1\_dropShadow\_2\_17" result="shape"></feBlend>

                                </filter>

                                <clipPath id="clip0\_2\_17">

                                    <rect width="28.0702" height="28.0702" fill="white" transform="translate(0.403503 0.526367)"></rect>

                                </clipPath>

                            </defs>

                        </svg>

                    </button>

                </div>

            </div>

            <section class="image-gallery">

                <div class="img-card"><img src="images/img-1.jpg" alt="image"></div>

                <div class="img-card"><img src="images/img-2.jpg" alt="image"></div>

                <div class="img-card"><img src="images/img-3.jpg" alt="image"></div>

                <div class="img-card"><img src="images/img-4.jpg" alt="image"></div>

            </section>

        </div>

    </div>

    <!-- Script JS -->

    <script>

        function generateImage() {

  const query = document.querySelector('.searchInput').value;

  const clientId = 'tiPGSSsgw9JBOq3x-dxdjpUKpApceJ7xqjHcua9bGGY';

  const url = `https://api.unsplash.com/photos/random?query=${query}&count=4&client\_id=${clientId}`;

  fetch(url)

  .then(response => response.json())

  .then(data => {

      const images = document.querySelectorAll('.image-gallery img');

      data.forEach((item, index) => {

          images[index].src = item.urls.regular;

          images[index].alt = item.alt\_description;

      });

  })

  .catch(error => console.error('Error fetching images:', error));

}

    </script>

</body>

</html>