```
<!DOCTYPE html>

<html>

<head>

    <title>Hellman Key Exchange</title>

</head>

<body>

    <h2>Alice's</h2>

    <label for="p">Enter prime number p: </label>

    <input type="text" id="p" /><br><br>


    <label for="g">Enter primitive root g: </label>

    <input type="text" id="g" /><br><br>


    <label for="x">Enter Alice's secret key x: </label>

    <input type="text" id="x" /><br><br>


    <button onclick="calculateA()">Calculate A</button><br><br>

    <label for="receivedA">Alice's calculated A: </label>

    <input type="text" id="receivedA" readonly /><br><br>





    <button onclick="calculateSecretKeySkA()">Secret key</button><br><br>

    <label for="secretKeyA">Shared secret key: </label>

    <input type="text" id="secretKeyA" readonly /><br><br>



    <h2>Bob's</h2>

    <label for="y">Enter Bob's secret key y: </label>

    <input type="text" id="y" /><br><br>
    <button onclick="calculateB()">Calculate B</button><br><br>
    <label for="calculatedB">Calculated B: </label>
    <input type="text" id="calculatedB" readonly /><br><br>

    <button onclick="calculateSecretKeySkB()">Secret key</button><br><br>
    <label for="secretKeyB">Shared secret key: </label>
    <input type="text" id="secretKeyB" readonly /><br><br>

    <script>
        let p, g, x, y, A, B, secretKey;

function isPrime(n) {
    if (n <= 1) {
        return false;
    }
    if (n <= 3) {
        return true;
    }
    if (n % 2 == 0 || n % 3 == 0) {
        return false;
    }
    for (let i = 5; i * i <= n; i = i + 6) {
        if (n % i == 0 || n % (i + 2) == 0) {
            return false;
        }
    }
    return true;
}

function isPrimitiveRoot(g, p) {
    let phi = p - 1; // Euler's totient function
    let factors = [];
    for (let i = 2; i * i <= phi; i++) {
        if (phi % i === 0) {
            factors.push(i);
            while (phi % i === 0) {
                phi /= i;
            }
        }
    }
    if (phi > 1) {
        factors.push(phi);
    }
    for (let factor of factors) {
        if (Math.pow(g, (p - 1) / factor) % p === 1) {
            return false;
        }
    }
    return true;
}

function calculateA() {
    p = parseInt(document.getElementById("p").value);
    g = parseInt(document.getElementById("g").value);
    x = parseInt(document.getElementById("x").value);

    if (!isPrime(p)) {
        alert("Please enter a prime number for p.");
        return;
    }

    if (!isPrimitiveRoot(g, p)) {
        alert("Please enter a primitive root for p.");
        return;
    }

    A = (g ** x) % p;

    document.getElementById("receivedA").value = A;
}

function calculateB() {
    y = parseInt(document.getElementById("y").value);

    if (!isPrime(p)) {
        alert("Please enter a prime number for p.");
        return;
    }

    if (!isPrimitiveRoot(g, p)) {
        alert("Please enter a primitive root for p.");
        return;
    }

    B = (g ** y) % p;

    document.getElementById("calculatedB").value = B;

    calculateSecretKey();
}

function calculateSecretKeySkA() {
    secretKey = (B ** x) % p;
    document.getElementById("secretKeyA").value = secretKey;
}

function calculateSecretKeySkB() {
    secretKey = (A ** y) % p;

    document.getElementById("secretKeyB").value = secretKey;
}
    </script>
</body>
</html>
```