Documentation

To build a web page from scratch, you need two main components: the Frontend and the backend.

The frontend is the visual and interactive part, the user interface that visitors see and interact with. It's where design, layout, and user experience come together.

The backend, on the other hand, is the engine behind the scenes. It handles the logic, data processing, and communication with the server that make the website function properly.

In our case, the project is a photo selection and sending system. The user will first select their generation, and the site will display the photos associated with that group. Then, they'll be able to choose the images they want and have them sent directly to their email address.

Our frontend:
It's built with 4 main files.

```
Index.html
```

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Formulario de Generación</title>
  <link rel="stylesheet" href="style.css">
  <link href="https://fonts.googleapis.com/css2?
family=Montserrat:wght@100;200;300;400;600;700&display=swap"
rel="stylesheet">
  <script type="text/javascript">
  function googleTranslateElementInit() {
    new google.translate.TranslateElement({
      pageLanguage: 'es',
      includedLanguages: 'en,fr,de,it',
      layout: google.translate.TranslateElement.InlineLayout.SIMPLE
    }, 'google_translate_element');
  }
  </script>
<script
src="//translate.google.com/translate_a/element.jscb=googleTranslateElementI
nit"></script>
</head>

<body>
  <div class="contenedor">
```

```
  <header class="header">
    <h1 class="header-title">Generaciones CSM</h1>
  </header>
  <div id="google_translate_element"></div>
    <hr class="separador">
    <h1>Registro y envío</h1>
    <form id="formulario">
      <label for="email">Ingresa tu e-mail:</label>
      <input type="email" id="email" name="email"
placeholder="ejemplo@correo.com" required>
      <label for="generacion">Selecciona tu generación:</label>
      <input type="text" id="generacion" name="generacion" placeholder="Ej.
2020" required>
      <button type="submit" id="verBtn">Ver</button>
        <script>
          document.getElementById('formulario').addEventListener('submit',
function(event) {
            event.preventDefault();
            const generacion = document.getElementById('generacion').value;
            const email = document.getElementById('email').value;
            if (!generacion || !email) {
              alert('Por favor completa ambos campos.');
              return;
            }
 window.open(`selector.html?gen=${encodeURIComponent(generacion)}`,
'_blank');
          });
        </script>
    </form>
  </div>
  <canvas id="swisscross"></canvas>
  <script src="main.js"></script>
</body>
</html>
```

style.css

```
body {
  font-family: 'Montserrat', sans-serif;
  font-weight: 100;
  background: #f0f0f0;
  color: #333;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
}
```

```css
.contenedor {
  background: white;
  padding: 30px;
  border-radius: 15px;
  box-shadow: 0 0 15px rgba(0,0,0,0.1);
  width: 350px;
}

h1 {
  font-weight: 550;
  text-align: center;
  margin-bottom: 25px;
  margin-top: 25px;
}

input, textarea, select {
  font-family: 'Montserrat', sans-serif;
  font-weight: 300;
  box-sizing: border-box;
}

form label {
  display: block;
  margin-top: 15px;
  font-weight: 500;
}

form input {
  width: 100%;
  padding: 12px;
  margin-top: 5px;
  border: 1px solid #ccc;
  border-radius: 8px;
  box-sizing: border-box;
}

form button {
  margin-top: 25px;
  width: 100%;
  padding: 12px;
  border: 1px solid #ccc;
  border-radius: 8px;
  background-color: #d90000;
  color: white;
  font-size: 16px;
  cursor: pointer;
  box-sizing: border-box;
}

form button:hover {
```

```css
  background-color: #b20000;
}

* {
  margin: 0;
  padding: 0;
  font-family: "Montserrat", sans-serif;
  font-display: swap;
}

.header {
  width: 100%;
  background-color: white;
  display: flex;
  justify-content: space-between;
  align-items: center;
  padding: 5px;
  box-sizing: border-box;
  margin-bottom: 10px;
}

.header-title {
  text-align: center;
  width: 100%;
  justify-content: center;
  display: flex;
  align-items: center;
  color: #E32236;
  font-size: 24px;
  font-weight: 300;
  margin: 0;
}

.form-wrapper {
  display: flex;
  justify-content: center;
  align-items: center;
  min-height: calc(100vh - 80px);
  padding-top: 30px;
}

.separador {
  border: none;
  height: 2px;
  background-color: #c5c5c5;
  margin: 20px;
  width: 90%;
  align-self: center;
}
```

```
#swisscross {
  position: fixed;
  bottom: 0;
  left: 0;
  width: 100%;
  height: 200px;
  z-index: -1;
  pointer-events: none;
}
```

Those last two files conform the user interface, where the user can input his e-mail, and generation. The `.css` file helps us with the design of the boxes, inputs, titles, etc. Morover, the `.html` file helps us to adapt the page to rezising, translating, asking for inputs among other things.

Additionally we have a `main.js` wich helps us with the decoration of our page. In this case we decided to generate swiss crosses procedurally. See the following code:

```
// Procedural Crosses
const canvas = document.getElementById('swisscross');
const ctx = canvas.getContext('2d');
let crosses = [];
const total = 100;

function resizeCanvas() {
  canvas.width = window.innerWidth;
  canvas.height = 200;
}

function createCross() {
  const x = Math.random() * canvas.width;
  const y = canvas.height + Math.random() * 100;
  const size = 5 + Math.random() * 20;
  const speed = 0.3 + Math.random() * 0.7;
  return { x, y, size, speed };
}

function initCrosses() {
  resizeCanvas();
  crosses = [];
  for (let i = 0; i < total; i++) {
    crosses.push(createCross());
  }
}

function drawCross(cross) {
  const opacity = 1 - (canvas.height - cross.y) / canvas.height;
```

```
  ctx.save();
  ctx.translate(cross.x, cross.y);
  ctx.strokeStyle = `rgba(255, 0, 0, ${opacity})`;
  ctx.lineWidth = Math.max(6, cross.size / 6);
  const s = cross.size / 2;
  ctx.beginPath();
  ctx.moveTo(-s, 0);
  ctx.lineTo(s, 0);
  ctx.moveTo(0, -s);
  ctx.lineTo(0, s);
  ctx.stroke();
  ctx.restore();
}

function animate() {
  ctx.clearRect(0, 0, canvas.width, canvas.height);
  for (let i = 0; i < crosses.length; i++) {
    let c = crosses[i];
    c.y -= c.speed;
    if (c.y < -c.size) {
      crosses[i] = createCross();
    }
    drawCross(c);
  }
  requestAnimationFrame(animate);
}
window.addEventListener('resize', () => {
  resizeCanvas();
  initCrosses();
});
initCrosses();
animate();
```