

Speedway Motorsport

Davi Portes – 10417139; Renato Ribeiro – 10410267; Paul Heinritz- 10425884.

Escolhemos para o foco de nosso projeto um website para todos os entusiastas mundo a fora, com opções de preparações, peças customizadas ou até carros por inteiro sendo preparados unicamente pela empresa, decidimos transmitir nossa paixão por motores, carros e velocidade em um meio onde todo entusiasta pode se encontrar e expressar seu estilo de vida em meio a sua máquina.

Nós escolhemos dar foco na parte a cultura japonesa, onde a customização tanto visual como a mecânica é uma cultura fortemente valorizada, e tentamos transmitir um pouco disso pelo nosso website para as pessoas que infelizmente não vivem em um lugar onde essa cultura seja fortemente agravada, portanto, nosso foco é além de fornecer tais opções de customizações.

O tipo de website escolhido foi retratar uma preparadora automotiva especializada em carros de origem japonesas, dando foco na parte de customização tanto visual como mecânica do veículo, com diversas abas com inúmeras opções de performance, seja para um modelo de carro específico ou até peças universais, dando a possibilidade do usuário se aventurar no mundo da customização, ou até mesmo a possibilidade de requisitar algo único feito especialmente para ele, ou até mesmo comprar um veículo por inteiro feito especialmente pela empresa, dando assim inúmeras possibilidades de escolha, assim como inúmeras opções de customização, visual, suspensão, algo voltado para o conforto ou até para um uso mais severo como uma pista de arrancada, ou um circuito de trackday em um kartódromo, assim valorizando a cultura da customização e influenciando o usuário a seguir o seu sonho e transmitir o seu sentimento e seu estilo diante ao seu carro.

Ajustes entrega 2:

Ajuste na Galeria (CSS Grid Responsivo)

css grid para ajustar as imagens independentemente do tamanho da tela

```
/* CSS Grid */
.image-container {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
  gap: 15px;
  justify-items: center;
  max-width: 900px;
  margin: auto;
}
```

Cabeçalho Fixo + transparência

Antes: O cabeçalho estava sobrepondo itens.

Agora: Não sobrepõe mais .

```
header {
  background-color: rgba(34, 34, 34, 0.9); /* 0.9 referente a transparencia */
  width: 100%;
  padding: 15px;
  position: fixed;
  top: 0;
  left: 0;
  z-index: 1000; /*trava o cabeçalho no topo */
  display: flex;
  justify-content: space-between;
  align-items: center;
  padding: 15px 30px;
  box-sizing: border-box;
}
```

Ajustes para Celulares

Antes: layout não se ajustava em telas pequenas.

Agora: Foi utilizado Media Query para otimizar o site versão mobile.

```

/* Media Query */
@media (max-width: 768px) {
  header {
    flex-direction: column;
    text-align: center;
  }

  nav ul {
    flex-wrap: nowrap; /* Não deixa o menu no mobile ficar empilhado */
    justify-content: center;
  }
}

```

Efeito de Hover (zoom) nas Imagens + padronização

Antes: Imagens estavam estáticas e cada uma com um tamanho.

Agora: Existe um efeito de zoom nas imagens ao passar o mouse e agora todas respeitam o mesmo tamanho.

```

.image-container img {
  width: 100%;
  max-width: 280px;
  height: 180px; /* Tamanho padrão nas imagens */
  object-fit: cover; /* Não estica as imagens */
  border-radius: 10px;
  transition: transform 0.3s; /* garante que o efeito de hover seja gradativo */
}

.image-container img:hover {
  transform: scale(1.05); /* Efeito de zoom ao passar o mouse */
}

```

Efeito de Hover (cor) no cabeçalho (web)

Antes: cabeçalho estático

Agora: Altera sua cor caso passe o mouse em cima.

```

nav ul li a:hover {
  color: #ffcc00;
}

```

Ajustes entrega 3:

Criação grid de produtos

Antes: Galeria de fotos

Agora: Transformamos a galeria de fotos em uma grid de produtos, contendo nome, preço e botão para adicionar ao carrinho de compras.

```
<section id="gallery" class="gallery">
  <h2>Produtos</h2>
  <div class="image-container">
    <div class="product">
      
      <p class="product-name">Turbo HKS</p>
      <p class="product-price">R$ 20.000,00</p>
      <button class="add-to-cart">Adicionar ao Carrinho</button>
    </div>
  </div>
```

Implementação da função “Carrinho de compras”

Antes: Não existia interação com os produtos

Agora: Possuímos a função de adicionar os produtos ao carrinho onde o mesmo soma os valores e é possível visualizar todos os produtos adicionados.

```

let cart = [];

const addToCartButtons = document.querySelectorAll('.add-to-cart');
const viewCartButton = document.getElementById('view-cart');

// somar os valores
function updateCartButton() {
  const total = cart.reduce((acc, item) => {
    const price = parseFloat(item.price.replace('R$', '').replace('.', '').replace(',', '.'));
    return acc + price;
  }, 0);
  viewCartButton.textContent = `Carrinho: R$ ${total.toFixed(2).replace('.', ',')}`;
}

// Adicionar produtos ao carrinho
addToCartButtons.forEach(button => {
  button.addEventListener('click', (event) => {
    const product = event.target.closest('.product');
    const productName = product.querySelector('.product-name').textContent;
    const productPrice = product.querySelector('.product-price').textContent;

    cart.push({ name: productName, price: productPrice });
    alert(`${productName} foi adicionado ao carrinho!`);
    updateCartButton(); // Atualiza o valor do carrinho
  });
});

```

```

// visualiza o carrinho
viewCartButton.addEventListener('click', () => {
  if (cart.length === 0) {
    alert("Seu carrinho está vazio.");
    return;
  }

  let cartDetails = "Produtos no Carrinho:\n\n";
  cart.forEach((item, index) => {
    cartDetails += `${index + 1}. ${item.name} - ${item.price}\n`;
  });

  const total = cart.reduce((acc, item) => {
    const price = parseFloat(item.price.replace('R$', '').replace('.', '').replace(',', '.'));
    return acc + price;
  }, 0);
  cartDetails += `\nTotal: R$ ${total.toFixed(2).replace('.', ',')}`;
  alert(cartDetails);
});

```