//About.jsx

import React, { useState, useEffect } from 'react';

import Modal from 'react-modal';

import { FaInfoCircle, FaLinkedin, FaGithub, FaEnvelope, FaArrowRight, FaArrowLeft } from 'react-icons/fa';

import { motion, AnimatePresence } from 'framer-motion';

import { Swiper, SwiperSlide } from 'swiper/react';

import { Navigation, Pagination, Autoplay, EffectCoverflow } from 'swiper/modules';

import 'swiper/css';

import 'swiper/css/navigation';

import 'swiper/css/pagination';

import 'swiper/css/effect-coverflow';

// Images d'arrière-plan et de décoration

import heroImage from '../assets/hero-bg.jpg';

import abstractShape1 from '../assets/abstract-1.svg';

import abstractShape2 from '../assets/abstract-2.svg';

// Images de l'équipe

import team1 from '../assets/team1.jfif';

import team2 from '../assets/team2.jfif';

import team3 from '../assets/team3.jfif';

// Images des projets

const projectImages = [

'/images/project1.jpg',

'/images/project2.jpg',

'/images/project3.jpg',

'/images/project4.jpg',

];

const customStyles = {

content: {

top: '50%',

left: '50%',

right: 'auto',

bottom: 'auto',

transform: 'translate(-50%, -50%)',

maxWidth: '30rem',

width: '90%',

borderRadius: '1.5rem',

padding: '2.5rem',

boxShadow: '0 20px 35px rgba(0,0,0,0.25)',

animation: 'fadeIn 0.4s ease-in-out',

border: 'none',

background: 'linear-gradient(145deg, #ffffff, #f8f9fa)',

},

overlay: {

backgroundColor: 'rgba(0, 0, 0, 0.85)',

backdropFilter: 'blur(8px)',

},

};

Modal.setAppElement('#root');

const About = () => {

const [modalIsOpen, setIsOpen] = useState(false);

const [selectedEngagement, setSelectedEngagement] = useState(null);

const [activeIndex, setActiveIndex] = useState(0);

const engagements = [

{

title: 'Transparence',

description: 'Nous communiquons ouvertement sur nos processus et résultats.',

icon: '🔍',

color: 'from-blue-400 to-blue-600'

},

{

title: 'Responsabilité',

description: 'Nous assumons pleinement la qualité de notre travail.',

icon: '⚡',

color: 'from-purple-400 to-purple-600'

},

{

title: 'Innovation',

description: 'Nous adoptons les dernières technologies pour des tests efficaces.',

icon: '💡',

color: 'from-yellow-400 to-orange-600'

},

{

title: 'Collaboration',

description: 'Nous travaillons main dans la main avec vos équipes.',

icon: '🤝',

color: 'from-green-400 to-green-600'

},

{

title: 'Éthique',

description: 'Nous respectons les normes les plus strictes en matière de confidentialité.',

icon: '🛡️',

color: 'from-red-400 to-red-600'

},

];

const teamMembers = [

{

name: 'Sarah L.',

title: 'Ingénieure QA Senior',

img: team1,

bio: "Plus de 8 ans d'expérience en tests automatisés",

expertise: ['Selenium', 'Cypress', 'Jest'],

social: {

linkedin: "#",

github: "#",

email: "sarah@example.com"

}

},

// Ajoutez les autres membres...

];

const statistics = [

{ number: '500+', label: 'Projets Réalisés' },

{ number: '98%', label: 'Clients Satisfaits' },

{ number: '24/7', label: 'Support Client' },

{ number: '50+', label: 'Experts QA' },

];

return (

<div className="font-sans overflow-hidden">

{/\* HERO SECTION avec Parallax et Animations \*/}

<section className="relative h-screen flex items-center justify-center overflow-hidden">

<motion.div

initial={{ scale: 1.2, opacity: 0 }}

animate={{ scale: 1, opacity: 1 }}

transition={{ duration: 1.5 }}

className="absolute inset-0 z-0"

>

<div

className="w-full h-full bg-cover bg-center bg-fixed"

style={{

backgroundImage: `url(${heroImage})`,

filter: 'brightness(0.7)'

}}

/>

</motion.div>

<div className="absolute inset-0 bg-gradient-to-b from-black/60 via-black/40 to-transparent" />

<motion.div

initial={{ y: 100, opacity: 0 }}

animate={{ y: 0, opacity: 1 }}

transition={{ delay: 0.5, duration: 1 }}

className="relative z-10 text-center px-4 max-w-5xl"

>

<h1 className="text-7xl font-extrabold mb-8 text-white">

<span className="bg-clip-text text-transparent bg-gradient-to-r from-blue-400 to-purple-600">

Excellence en Tests Logiciels

</span>

</h1>

<p className="text-2xl text-gray-200 mb-12 leading-relaxed">

Nous transformons la qualité logicielle en avantage compétitif

</p>

<motion.button

whileHover={{ scale: 1.05 }}

whileTap={{ scale: 0.95 }}

className="bg-gradient-to-r from-blue-500 to-purple-600 text-white px-8 py-4 rounded-full

text-xl font-semibold shadow-lg hover:shadow-2xl transition-all duration-300"

>

Découvrir Notre Expertise

</motion.button>

</motion.div>

{/\* Animated Scroll Indicator \*/}

<motion.div

animate={{ y: [0, 10, 0] }}

transition={{ repeat: Infinity, duration: 1.5 }}

className="absolute bottom-10 left-1/2 transform -translate-x-1/2"

>

<div className="w-6 h-10 border-2 border-white rounded-full flex justify-center">

<div className="w-2 h-2 bg-white rounded-full mt-2" />

</div>

</motion.div>

</section>

{/\* STATISTICS SECTION avec Animation au Scroll \*/}

<section className="py-20 bg-gradient-to-r from-gray-900 to-black text-white">

<div className="max-w-7xl mx-auto px-6">

<div className="grid grid-cols-1 md:grid-cols-4 gap-10">

{statistics.map((stat, index) => (

<motion.div

key={index}

initial={{ opacity: 0, y: 50 }}

whileInView={{ opacity: 1, y: 0 }}

viewport={{ once: true }}

transition={{ delay: index \* 0.1 }}

className="text-center"

>

<h3 className="text-5xl font-bold bg-gradient-to-r from-blue-400 to-purple-600

bg-clip-text text-transparent mb-4">

{stat.number}

</h3>

<p className="text-xl text-gray-300">{stat.label}</p>

</motion.div>

))}

</div>

</div>

</section>

{/\* SERVICES CAROUSEL \*/}

<section className="py-24 bg-gradient-to-b from-gray-100 to-white">

<div className="max-w-7xl mx-auto px-6">

<motion.div

initial={{ opacity: 0, y: 50 }}

whileInView={{ opacity: 1, y: 0 }}

viewport={{ once: true }}

className="text-center mb-16"

>

<h2 className="text-5xl font-bold text-gray-800 mb-6">Nos Services</h2>

<p className="text-xl text-gray-600">Des solutions complètes pour vos besoins en QA</p>

</motion.div>

<Swiper

modules={[Navigation, Pagination, Autoplay, EffectCoverflow]}

effect="coverflow"

grabCursor={true}

centeredSlides={true}

slidesPerView={3}

coverflowEffect={{

rotate: 50,

stretch: 0,

depth: 100,

modifier: 1,

slideShadows: true,

}}

autoplay={{

delay: 3000,

disableOnInteraction: false,

}}

pagination={{ clickable: true }}

navigation

className="mySwiper"

>

{engagements.map((service, index) => (

<SwiperSlide key={index}>

<div className={`bg-gradient-to-br ${service.color} p-8 rounded-2xl text-white

transform hover:scale-105 transition-transform duration-300`}>

<div className="text-5xl mb-6">{service.icon}</div>

<h3 className="text-2xl font-bold mb-4">{service.title}</h3>

<p className="text-gray-100">{service.description}</p>

</div>

</SwiperSlide>

))}

</Swiper>

</div>

</section>

{/\* TEAM SECTION avec Hover Effects \*/}

<section className="py-24 bg-gradient-to-br from-gray-900 via-gray-800 to-black text-white">

<div className="max-w-7xl mx-auto px-6">

<motion.div

initial={{ opacity: 0, y: 50 }}

whileInView={{ opacity: 1, y: 0 }}

viewport={{ once: true }}

className="text-center mb-16"

>

<h2 className="text-5xl font-bold mb-6">Notre Équipe d'Experts</h2>

<p className="text-xl text-gray-300">Des professionnels passionnés par la qualité</p>

</motion.div>

<div className="grid md:grid-cols-3 gap-12">

{teamMembers.map((member, idx) => (

<motion.div

key={idx}

initial={{ opacity: 0, scale: 0.9 }}

whileInView={{ opacity: 1, scale: 1 }}

viewport={{ once: true }}

whileHover={{ y: -10 }}

className="relative group"

>

<div className="absolute inset-0 bg-gradient-to-r from-blue-600 to-purple-600

rounded-2xl transform rotate-6 group-hover:rotate-12 transition-transform duration-300" />

<div className="relative bg-gray-800 p-8 rounded-2xl">

<div className="relative w-40 h-40 mx-auto mb-6">

<img

src={member.img}

alt={member.name}

className="w-full h-full rounded-full object-cover"

/>

<div className="absolute inset-0 rounded-full bg-gradient-to-r from-blue-400 to-purple-600

opacity-0 group-hover:opacity-30 transition-opacity duration-300" />

</div>

<h3 className="text-2xl font-bold mb-2">{member.name}</h3>

<p className="text-blue-400 mb-4">{member.title}</p>

<div className="flex flex-wrap gap-2 mb-6">

{member.expertise.map((skill, i) => (

<span key={i} className="px-3 py-1 bg-gray-700 rounded-full text-sm">

{skill}

</span>

))}

</div>

<div className="flex justify-center gap-4">

{Object.entries(member.social).map(([platform, link], i) => (

<a

key={i}

href={link}

className="text-gray-400 hover:text-blue-400 transition-colors duration-300"

>

{platform === 'linkedin' && <FaLinkedin size={24} />}

{platform === 'github' && <FaGithub size={24} />}

{platform === 'email' && <FaEnvelope size={24} />}

</a>

))}

</div>

</div>

</motion.div>

))}

</div>

</div>

</section>

{/\* PROJECTS SHOWCASE \*/}

<section className="py-24 bg-white">

<div className="max-w-7xl mx-auto px-6">

<motion.div

initial={{ opacity: 0, y: 50 }}

whileInView={{ opacity: 1, y: 0 }}

viewport={{ once: true }}

className="text-center mb-16"

>

<h2 className="text-5xl font-bold text-gray-800 mb-6">Nos Réalisations</h2>

<p className="text-xl text-gray-600">Découvrez nos projets les plus marquants</p>

</motion.div>

<Swiper

modules={[Navigation, Pagination, Autoplay]}

spaceBetween={30}

slidesPerView={1}

navigation

pagination={{ clickable: true }}

autoplay={{

delay: 5000,

disableOnInteraction: false,

}}

breakpoints={{

640: {

slidesPerView: 2,

},

1024: {

slidesPerView: 3,

},

}}

>

{projectImages.map((image, index) => (

<SwiperSlide key={index}>

<motion.div

whileHover={{ scale: 1.05 }}

className="relative overflow-hidden rounded-2xl shadow-xl"

>

<img

src={image}

alt={`Project ${index + 1}`}

className="w-full h-64 object-cover"

/>

<div className="absolute inset-0 bg-gradient-to-t from-black/80 to-transparent

opacity-0 hover:opacity-100 transition-opacity duration-300

flex items-end p-6">

<div>

<h3 className="text-xl font-bold text-white mb-2">

Projet {index + 1}

</h3>

<p className="text-gray-200">

Description du projet avec ses spécificités

</p>

</div>

</div>

</motion.div>

</SwiperSlide>

))}

</Swiper>

</div>

</section>

{/\* CTA SECTION avec Animation \*/}

<section className="relative py-24 overflow-hidden">

<div className="absolute inset-0 bg-gradient-to-r from-blue-600 to-purple-700" />

<div className="absolute inset-0 opacity-20">

<div className="absolute top-0 left-0 w-40 h-40 bg-white rounded-full blur-3xl transform -translate-x-1/2 -translate-y-1/2" />

<div className="absolute bottom-0 right-0 w-40 h-40 bg-purple-300 rounded-full blur-3xl transform translate-x-1/2 translate-y-1/2" />

</div>

<motion.div

initial={{ opacity: 0, y: 50 }}

whileInView={{ opacity: 1, y: 0 }}

viewport={{ once: true }}

className="relative max-w-4xl mx-auto px-6 text-center"

>

<h2 className="text-4xl font-bold text-white mb-8">

Prêt à Transformer Votre Approche QA ?

</h2>

<p className="text-xl text-blue-100 mb-12">

Rejoignez les entreprises qui font confiance à notre expertise pour garantir

la qualité de leurs applications

</p>

<motion.button

whileHover={{ scale: 1.05 }}

whileTap={{ scale: 0.95 }}

className="bg-white text-blue-600 text-lg font-semibold px-8 py-4

rounded-full shadow-lg hover:shadow-2xl transform transition-all duration-300"

>

Commencer Maintenant

</motion.button>

</motion.div>

</section>

</div>

);

};

export default About;

//Footer.jsx

import React from 'react';

import { motion } from 'framer-motion';

import { FaFacebookF, FaTwitter, FaInstagram, FaLinkedinIn, FaHeart } from 'react-icons/fa';

import { Link } from 'react-router-dom';

const Footer = () => {

const footerLinks = {

services: [

{ name: 'Tests Automatisés', href: '#' },

{ name: 'Tests Manuels', href: '#' },

{ name: 'Tests de Performance', href: '#' },

{ name: 'Tests de Sécurité', href: '#' },

],

company: [

{ name: 'À propos', href: '/about' },

{ name: 'Carrières', href: '#' },

{ name: 'Blog', href: '#' },

{ name: 'Contact', href: '/contact' },

],

resources: [

{ name: 'Documentation', href: '#' },

{ name: 'Guides', href: '#' },

{ name: 'Webinaires', href: '#' },

{ name: 'API', href: '#' },

],

legal: [

{ name: 'Mentions légales', href: '#' },

{ name: 'Confidentialité', href: '#' },

{ name: 'CGU', href: '#' },

],

};

const socialLinks = [

{ icon: <FaFacebookF />, href: '#' },

{ icon: <FaTwitter />, href: '#' },

{ icon: <FaInstagram />, href: '#' },

{ icon: <FaLinkedinIn />, href: '#' },

];

return (

<footer className="bg-gray-900 text-white relative overflow-hidden">

{/\* Gradient Overlay \*/}

<div className="absolute inset-0 bg-gradient-to-b from-gray-900/50 to-gray-900" />

{/\* Main Footer Content \*/}

<div className="relative max-w-7xl mx-auto px-4 py-16">

<div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-5 gap-12">

{/\* Brand Column \*/}

<div className="lg:col-span-2">

<motion.div

initial={{ opacity: 0, y: 20 }}

whileInView={{ opacity: 1, y: 0 }}

transition={{ duration: 0.5 }}

>

<h2 className="text-3xl font-bold mb-6 bg-gradient-to-r from-blue-400 to-purple-600 bg-clip-text text-transparent">

QA Testing

</h2>

<p className="text-gray-400 mb-8 max-w-md">

Nous fournissons des solutions de test complètes pour garantir la qualité

de vos applications. Notre expertise fait la différence.

</p>

<div className="flex space-x-4">

{socialLinks.map((social, index) => (

<motion.a

key={index}

href={social.href}

whileHover={{ scale: 1.1, y: -3 }}

className="w-10 h-10 rounded-full bg-gray-800 flex items-center justify-center

text-gray-400 hover:bg-gradient-to-r hover:from-blue-500 hover:to-purple-600

hover:text-white transition-all duration-300"

>

{social.icon}

</motion.a>

))}

</div>

</motion.div>

</div>

{/\* Links Columns \*/}

{Object.entries(footerLinks).map(([category, links], idx) => (

<motion.div

key={category}

initial={{ opacity: 0, y: 20 }}

whileInView={{ opacity: 1, y: 0 }}

transition={{ duration: 0.5, delay: idx \* 0.1 }}

>

<h3 className="text-lg font-semibold mb-4 text-gray-100 capitalize">

{category}

</h3>

<ul className="space-y-3">

{links.map((link, index) => (

<li key={index}>

<Link

to={link.href}

className="text-gray-400 hover:text-blue-400 transition-colors duration-300"

>

{link.name}

</Link>

</li>

))}

</ul>

</motion.div>

))}

</div>

{/\* Newsletter Section \*/}

<motion.div

initial={{ opacity: 0, y: 20 }}

whileInView={{ opacity: 1, y: 0 }}

transition={{ duration: 0.5, delay: 0.4 }}

className="mt-16 pt-8 border-t border-gray-800"

>

<div className="max-w-md">

<h3 className="text-xl font-semibold mb-4">Restez informé</h3>

<p className="text-gray-400 mb-4">

Abonnez-vous à notre newsletter pour recevoir nos dernières actualités.

</p>

<form className="flex gap-2">

<input

type="email"

placeholder="Votre email"

className="flex-1 px-4 py-2 rounded-lg bg-gray-800 border border-gray-700

focus:outline-none focus:border-blue-500 text-gray-300"

/>

<motion.button

whileHover={{ scale: 1.02 }}

whileTap={{ scale: 0.98 }}

className="px-6 py-2 rounded-lg bg-gradient-to-r from-blue-500 to-purple-600

text-white font-medium hover:shadow-lg transition-all duration-300"

>

S'abonner

</motion.button>

</form>

</div>

</motion.div>

{/\* Bottom Bar \*/}

<motion.div

initial={{ opacity: 0 }}

whileInView={{ opacity: 1 }}

transition={{ duration: 0.5, delay: 0.6 }}

className="mt-16 pt-8 border-t border-gray-800 text-center text-gray-400"

>

<p className="flex items-center justify-center gap-2">

Made with <FaHeart className="text-red-500" /> by QA Testing Team

</p>

<p className="mt-2">

© {new Date().getFullYear()} QA Testing. Tous droits réservés.

</p>

</motion.div>

</div>

</footer>

);

};

export default Footer;

// LandingPage.jsx

import React, { useState, useEffect } from 'react';

import { motion, AnimatePresence } from 'framer-motion';

import image2 from "../assets/bk-footer.png";

import image from "../assets/st.jfif";

import image1 from "../assets/RobotLogo.png";

import ChatbotIcon from './chatbotIcon';

import { FaBrain, FaFile, FaComments, FaChartLine, FaBell, FaArrowRight, FaTimes } from "react-icons/fa";

import { Link } from 'react-router-dom';

// Flash Info Data

const flashNews = [

{

id: 1,

type: 'important',

title: "Nouvelle mise à jour majeure",

content: "Version 2.0 disponible avec de nouvelles fonctionnalités d'automatisation",

date: "2024-03-20"

},

{

id: 2,

type: 'event',

title: "Webinaire : Tests automatisés",

content: "Rejoignez-nous le 25 mars pour un webinaire sur les meilleures pratiques",

date: "2024-03-25"

},

{

id: 3,

type: 'update',

title: "Maintenance planifiée",

content: "Une maintenance est prévue le 30 mars de 2h à 4h",

date: "2024-03-30"

}

];

// Existing data structures...

const services = [/\* ... votre code existant ... \*/];

const avantages = [/\* ... votre code existant ... \*/];

const articles = [/\* ... votre code existant ... \*/];

const LandingPage = () => {

const [activeFlashNews, setActiveFlashNews] = useState(0);

const [showFlashNews, setShowFlashNews] = useState(true);

useEffect(() => {

const timer = setInterval(() => {

setActiveFlashNews((prev) => (prev + 1) % flashNews.length);

}, 5000);

return () => clearInterval(timer);

}, []);

return (

<div className="relative">

{/\* Flash Info Banner \*/}

<AnimatePresence>

{showFlashNews && (

<motion.div

initial={{ height: 0, opacity: 0 }}

animate={{ height: "auto", opacity: 1 }}

exit={{ height: 0, opacity: 0 }}

className="relative bg-gradient-to-r from-orange-600 to-orange-800 text-white"

>

<div className="max-w-7xl mx-auto px-4 py-3">

<div className="flex items-center justify-between">

<div className="flex items-center">

<FaBell className="h-5 w-5 mr-2 animate-bounce" />

<div className="flex flex-col">

<p className="font-semibold">

{flashNews[activeFlashNews].title}

</p>

<p className="text-sm text-orange-100">

{flashNews[activeFlashNews].content}

</p>

</div>

</div>

<div className="flex items-center space-x-4">

<button

onClick={() => setActiveFlashNews((prev) => (prev + 1) % flashNews.length)}

className="text-orange-100 hover:text-white transition-colors"

>

<FaArrowRight className="h-5 w-5" />

</button>

<button

onClick={() => setShowFlashNews(false)}

className="text-orange-100 hover:text-white transition-colors"

>

<FaTimes className="h-5 w-5" />

</button>

</div>

</div>

</div>

{/\* Progress bar \*/}

<div className="absolute bottom-0 left-0 h-1 bg-orange-300/30 w-full">

<motion.div

initial={{ width: "0%" }}

animate={{ width: "100%" }}

transition={{ duration: 5, repeat: Infinity }}

className="h-full bg-white"

/>

</div>

</motion.div>

)}

</AnimatePresence>

{/\* Hero Section - Amélioré \*/}

<section className="relative min-h-screen bg-cover bg-center bg-fixed" style={{ backgroundImage: `url(${image2})` }}>

<div className="absolute inset-0 bg-gradient-to-b from-black/70 via-black/50 to-black/70 flex items-center">

<motion.div

initial={{ opacity: 0, y: 20 }}

animate={{ opacity: 1, y: 0 }}

transition={{ duration: 0.8 }}

className="container mx-auto px-4 py-16 md:py-24 text-white"

>

<div className="md:w-1/2">

<h1 className="text-4xl md:text-5xl lg:text-6xl font-bold mb-6 leading-tight">

Simplifiez la gestion des tests logiciels avec

<span className="bg-gradient-to-r from-orange-400 to-orange-600 bg-clip-text text-transparent">

{" "}l'automatisation et l'IA

</span>

</h1>

<p className="text-xl mb-8 text-gray-300">

Créez, suivez et analysez vos tests en quelques clics grâce à une interface intuitive,

un chatbot intelligent et une base de connaissances complète.

</p>

<div className="flex flex-col sm:flex-row space-y-4 sm:space-y-0 sm:space-x-6">

<motion.button

whileHover={{ scale: 1.05 }}

whileTap={{ scale: 0.95 }}

className="bg-gradient-to-r from-orange-600 to-orange-700 text-white px-8 py-4

rounded-full font-medium shadow-lg hover:shadow-orange-500/20

transition duration-300"

>

Commencer gratuitement

</motion.button>

<motion.button

whileHover={{ scale: 1.05 }}

whileTap={{ scale: 0.95 }}

className="bg-white/10 backdrop-blur-sm text-white border-2 border-white/30

hover:bg-white/20 px-8 py-4 rounded-full font-medium transition duration-300"

>

Voir la démo

</motion.button>

</div>

</div>

</motion.div>

</div>

{/\* Animated scroll indicator \*/}

<motion.div

animate={{ y: [0, 10, 0] }}

transition={{ repeat: Infinity, duration: 1.5 }}

className="absolute bottom-10 left-1/2 transform -translate-x-1/2"

>

<div className="w-6 h-10 border-2 border-white/50 rounded-full flex justify-center">

<div className="w-2 h-2 bg-white rounded-full mt-2 animate-bounce" />

</div>

</motion.div>

</section>

{/\* Services Section - Amélioré \*/}

<section className="py-20 bg-gradient-to-b from-white to-gray-50">

<div className="max-w-7xl mx-auto px-4">

<motion.div

initial={{ opacity: 0, y: 20 }}

whileInView={{ opacity: 1, y: 0 }}

viewport={{ once: true }}

transition={{ duration: 0.6 }}

className="text-center mb-16"

>

<h2 className="text-4xl font-bold mb-6">Fonctionnalités clés</h2>

<p className="text-xl text-gray-600 max-w-3xl mx-auto">

Découvrez nos outils puissants pour optimiser votre processus de test

</p>

</motion.div>

<div className="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-4 gap-8">

{services.map((service, index) => {

const Icon = service.icon;

return (

<motion.div

key={index}

initial={{ opacity: 0, y: 20 }}

whileInView={{ opacity: 1, y: 0 }}

viewport={{ once: true }}

transition={{ duration: 0.5, delay: index \* 0.1 }}

whileHover={{ y: -10 }}

className="bg-white p-8 rounded-2xl shadow-lg hover:shadow-2xl

transition-all duration-300 group"

>

<div className="mb-6">

<Icon className="text-5xl text-orange-500 group-hover:text-orange-600

transition-colors duration-300" />

</div>

<h3 className="text-xl font-semibold mb-4">{service.title}</h3>

<p className="text-gray-600">{service.description}</p>

</motion.div>

);

})}

</div>

</div>

</section>

{/\* Avantages Section - Amélioré \*/}

<section className="py-20 bg-gray-900 text-white">

<div className="max-w-7xl mx-auto px-4">

<motion.div

initial={{ opacity: 0, y: 20 }}

whileInView={{ opacity: 1, y: 0 }}

viewport={{ once: true }}

transition={{ duration: 0.6 }}

className="text-center mb-16"

>

<h2 className="text-4xl font-bold mb-6">Vos avantages</h2>

<p className="text-xl text-gray-400 max-w-3xl mx-auto">

Optimisez votre processus de test et gagnez en efficacité

</p>

</motion.div>

<div className="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-4 gap-8">

{avantages.map((item, idx) => (

<motion.div

key={idx}

initial={{ opacity: 0, y: 20 }}

whileInView={{ opacity: 1, y: 0 }}

viewport={{ once: true }}

transition={{ duration: 0.5, delay: idx \* 0.1 }}

className="bg-gray-800 p-8 rounded-2xl hover:bg-gray-700

transition-all duration-300 transform hover:-translate-y-2"

>

<h3 className="text-xl font-semibold mb-4 text-orange-400">

{item.title}

</h3>

<p className="text-gray-300">{item.description}</p>

</motion.div>

))}

</div>

</div>

</section>

{/\* Articles Section - Amélioré \*/}

<section className="py-20 bg-white">

<div className="max-w-7xl mx-auto px-4">

<motion.div

initial={{ opacity: 0, y: 20 }}

whileInView={{ opacity: 1, y: 0 }}

viewport={{ once: true }}

transition={{ duration: 0.6 }}

className="text-center mb-16"

>

<h2 className="text-4xl font-bold mb-6">Articles proposés</h2>

<p className="text-xl text-gray-600 max-w-3xl mx-auto">

Restez à jour avec nos derniers articles et guides

</p>

</motion.div>

<div className="grid grid-cols-1 md:grid-cols-3 gap-8">

{articles.map((article, idx) => (

<motion.div

key={idx}

initial={{ opacity: 0, y: 20 }}

whileInView={{ opacity: 1, y: 0 }}

viewport={{ once: true }}

transition={{ duration: 0.5, delay: idx \* 0.1 }}

className="group"

>

<div className="bg-white rounded-2xl overflow-hidden shadow-lg

hover:shadow-2xl transition-all duration-300">

<div className="relative overflow-hidden">

<img

src={article.image}

alt={article.title}

className="w-full h-48 object-cover transform group-hover:scale-110

transition-transform duration-300"

/>

<div className="absolute inset-0 bg-gradient-to-t from-black/60 to-transparent" />

</div>

<div className="p-6">

<h3 className="text-xl font-semibold mb-3">{article.title}</h3>

<p className="text-gray-600 mb-4">{article.summary}</p>

<Link

to={`/article/${article.title.toLowerCase().replace(/ /g, '-')}`}

className="inline-flex items-center text-orange-600 hover:text-orange-700

font-medium transition-colors duration-300"

>

Lire l'article

<FaArrowRight className="ml-2 text-sm" />

</Link>

</div>

</div>

</motion.div>

))}

</div>

</div>

</section>

</div>

);

};

export default LandingPage;

//Chatbot

import React, { useState, useEffect } from 'react';

import ChatBot from 'react-simple-chatbot';

import axios from 'axios';

const ChatAPI = ({ steps, triggerNextStep }) => {

useEffect(() => {

const fetchData = async () => {

const userMessage = steps.userInput.value;

console.log('Message envoyé:', userMessage); // Debug

try {

const response = await axios.post(

'https://fictional-broccoli-g944v47xg6929x5x-5000.app.github.dev/ask',

{ question: userMessage },

{

headers: {

'X-Github-Token': 'ghu\_U2Q2Ps9TF5EGh1IVtEA7TFy4pew35s3BgQTZ',

'Content-Type': 'application/json',

},

}

);

console.log('Réponse complète de l\'API:', response); // Debug complet

console.log('Données de réponse:', response.data); // Debug données

// Vérification et traitement de la réponse

let botResponse = '';

if (response.data) {

if (typeof response.data === 'string') {

botResponse = response.data;

} else if (response.data.reply) {

botResponse = response.data.reply;

} else if (response.data.response) {

botResponse = response.data.response;

} else {

botResponse = JSON.stringify(response.data);

}

}

console.log('Réponse formatée:', botResponse); // Debug réponse finale

// Déclencher la prochaine étape avec la réponse

triggerNextStep({

value: botResponse,

trigger: 'askAgain'

});

} catch (error) {

console.error('Erreur détaillée:', error.response || error); // Debug erreur

triggerNextStep({

value: "Je suis désolé, je n'ai pas pu traiter votre demande. Pouvez-vous réessayer ?",

trigger: 'askAgain'

});

}

};

fetchData();

}, [steps, triggerNextStep]);

return (

<div style={{ padding: '10px', fontStyle: 'italic' }}>

Traitement de votre demande...

</div>

);

};

const steps = [

{

id: '1',

message: 'Bonjour, je suis votre assistant QA. Comment puis-je vous aider ?',

trigger: 'userInput',

},

{

id: 'userInput',

user: true,

trigger: 'processing',

},

{

id: 'processing',

component: <ChatAPI />,

waitAction: true,

asMessage: true,

},

{

id: 'askAgain',

message: 'Avez-vous une autre question ?',

trigger: 'userChoice',

},

{

id: 'userChoice',

options: [

{ value: 1, label: 'Oui', trigger: 'userInput' },

{ value: 2, label: 'Non, merci', trigger: 'end' },

],

},

{

id: 'end',

message: 'Merci d\'avoir utilisé notre assistant ! À bientôt !',

end: true,

},

];

const theme = {

background: '#f5f8fb',

fontFamily: 'Arial, sans-serif',

headerBgColor: '#E65100',

headerFontColor: '#fff',

headerFontSize: '16px',

botBubbleColor: '#E65100',

botFontColor: '#fff',

userBubbleColor: '#fff',

userFontColor: '#4a4a4a',

};

const config = {

floating: true,

width: "400px",

height: "500px",

placeholder: 'Tapez votre message...',

headerTitle: 'Assistant QA',

botAvatar: "🤖",

userAvatar: "👤",

floatingStyle: {

bottom: '20px',

right: '20px',

width: '400px',

height: '500px',

},

};

// Dans votre composant App

const App = () => {

return (

<BrowserRouter>

<div>

<AuthProvider>

<Navbar/>

<ThemeProvider theme={theme}>

<div className="chatbot-container">

<ChatBot

steps={steps}

{...config}

handleEnd={({ renderedSteps }) => {

console.log('Conversation terminée:', renderedSteps);

}}

cache={false}

enableMobileAutoFocus={true}

enableSmoothScroll={true}

botDelay={1000}

customDelay={1000}

userDelay={200}

recognitionEnable={false}

speechSynthesis={{ enable: false }}

/>

</div>

</ThemeProvider>

<Routes>

{/\* Vos routes existantes \*/}

</Routes>

</AuthProvider>

</div>

</BrowserRouter>

);

};

// Ajoutez ces styles dans votre fichier CSS

const styles = `

.chatbot-container {

position: fixed;

bottom: 20px;

right: 20px;

z-index: 9999;

}

.rsc-container {

box-shadow: 0 5px 20px rgba(0, 0, 0, 0.15) !important;

border-radius: 10px !important;

}

.rsc-header {

padding: 15px !important;

}

.rsc-content {

height: calc(100% - 85px) !important;

}

.rsc-ts-bubble {

padding: 12px 16px !important;

border-radius: 15px !important;

max-width: 85% !important;

margin: 8px 0 !important;

}

.rsc-ts-user {

background-color: #E3F2FD !important;

color: #1976D2 !important;

}

.rsc-ts-bot {

background-color: #E65100 !important;

color: white !important;

}

.rsc-footer {

border-top: 1px solid #eee !important;

}

.rsc-input {

padding: 12px !important;

font-size: 14px !important;

}

.loading-message {

padding: 10px;

color: #666;

font-style: italic;

text-align: center;

}

`;

export default App;

//Tables.jsx

import React, { useState, useEffect } from 'react';

import axios from 'axios';

import { Chart } from 'react-chartjs-2';

import 'chart.js/auto';

import {

FaUsers, FaClipboardList, FaTasks, FaChartLine,

FaSearch, FaBell, FaUserCircle, FaCog, FaSignOutAlt

} from 'react-icons/fa';

import { motion } from 'framer-motion';

const Tables = () => {

// États

const [activeTab, setActiveTab] = useState('dashboard');

const [demandes, setDemandes] = useState([]);

const [users, setUsers] = useState([]);

const [searchTerm, setSearchTerm] = useState('');

const [notifications, setNotifications] = useState([]);

const [selectedRessources, setSelectedRessources] = useState({});

const [stats, setStats] = useState({

totalDemandes: 0,

demandesEnCours: 0,

demandesTerminees: 0,

totalUsers: 0

});

// Chargement des données

useEffect(() => {

const fetchData = async () => {

try {

const [demandesRes, usersRes] = await Promise.all([

axios.get('http://localhost:8084/api/demandes'),

axios.get('http://localhost:8084/api/users')

]);

setDemandes(demandesRes.data);

setUsers(usersRes.data || mockUsers); // Utilisez les données mockées si l'API n'existe pas

updateStats(demandesRes.data, usersRes.data || mockUsers);

} catch (error) {

console.error('Erreur lors du chargement des données:', error);

}

};

fetchData();

}, []);

// Données mockées pour les utilisateurs

const mockUsers = [

{ id: 1, name: 'John Doe', email: 'john@example.com', role: 'Admin', status: 'Active' },

{ id: 2, name: 'Jane Smith', email: 'jane@example.com', role: 'User', status: 'Active' },

// Ajoutez plus d'utilisateurs si nécessaire

];

// Mise à jour des statistiques

const updateStats = (demandes, users) => {

setStats({

totalDemandes: demandes.length,

demandesEnCours: demandes.filter(d => d.statut === 'EN COURS').length,

demandesTerminees: demandes.filter(d => d.statut === 'TRAITEE').length,

totalUsers: users.length

});

};

// Filtrage des demandes

const filteredDemandes = demandes.filter(demande =>

demande.projet.toLowerCase().includes(searchTerm.toLowerCase()) ||

demande.sujet.toLowerCase().includes(searchTerm.toLowerCase())

);

// Données pour les graphiques

const chartData = {

// ... votre code existant pour chartData

};

const chartData2 = {

// ... votre code existant pour chartData2

};

return (

<div className="flex h-screen bg-gray-100">

{/\* Sidebar \*/}

<motion.div

initial={{ x: -250 }}

animate={{ x: 0 }}

className="w-64 bg-gray-900 text-white"

>

<div className="p-4">

<h2 className="text-2xl font-bold mb-8">Admin Panel</h2>

<nav className="space-y-2">

<SidebarItem

icon={<FaChartLine />}

text="Dashboard"

active={activeTab === 'dashboard'}

onClick={() => setActiveTab('dashboard')}

/>

<SidebarItem

icon={<FaClipboardList />}

text="Demandes"

active={activeTab === 'demandes'}

onClick={() => setActiveTab('demandes')}

/>

<SidebarItem

icon={<FaUsers />}

text="Utilisateurs"

active={activeTab === 'users'}

onClick={() => setActiveTab('users')}

/>

</nav>

</div>

</motion.div>

{/\* Main Content \*/}

<div className="flex-1 overflow-hidden">

{/\* Header \*/}

<header className="bg-white shadow-sm">

<div className="flex items-center justify-between px-6 py-4">

<div className="flex items-center space-x-4">

<h1 className="text-2xl font-semibold text-gray-800">

{activeTab === 'dashboard' && 'Dashboard'}

{activeTab === 'demandes' && 'Gestion des Demandes'}

{activeTab === 'users' && 'Gestion des Utilisateurs'}

</h1>

</div>

<div className="flex items-center space-x-4">

<SearchBar value={searchTerm} onChange={setSearchTerm} />

<NotificationBell notifications={notifications} />

<UserMenu />

</div>

</div>

</header>

{/\* Main Content Area \*/}

<main className="p-6 overflow-auto h-[calc(100vh-4rem)]">

{activeTab === 'dashboard' && (

<DashboardView stats={stats} chartData={chartData} chartData2={chartData2} />

)}

{activeTab === 'demandes' && (

<DemandesView

demandes={filteredDemandes}

handleStatusChange={handleStatusChange}

handleAffecter={handleAffecter}

selectedRessources={selectedRessources}

setSelectedRessources={setSelectedRessources}

/>

)}

{activeTab === 'users' && (

<UsersView users={users} />

)}

</main>

</div>

</div>

);

};

// Composants auxiliaires

const SidebarItem = ({ icon, text, active, onClick }) => (

<motion.button

whileHover={{ scale: 1.02 }}

whileTap={{ scale: 0.98 }}

className={`flex items-center space-x-3 w-full px-4 py-3 rounded-lg transition-colors

${active ? 'bg-blue-600' : 'hover:bg-gray-800'}`}

onClick={onClick}

>

{icon}

<span>{text}</span>

</motion.button>

);

const SearchBar = ({ value, onChange }) => (

<div className="relative">

<FaSearch className="absolute left-3 top-1/2 transform -translate-y-1/2 text-gray-400" />

<input

type="text"

placeholder="Rechercher..."

value={value}

onChange={(e) => onChange(e.target.value)}

className="pl-10 pr-4 py-2 rounded-lg border border-gray-300 focus:outline-none focus:ring-2 focus:ring-blue-500"

/>

</div>

);

const NotificationBell = ({ notifications }) => (

<motion.button

whileHover={{ scale: 1.1 }}

whileTap={{ scale: 0.9 }}

className="relative p-2"

>

<FaBell className="text-gray-600 text-xl" />

{notifications.length > 0 && (

<span className="absolute top-0 right-0 bg-red-500 text-white text-xs rounded-full w-5 h-5 flex items-center justify-center">

{notifications.length}

</span>

)}

</motion.button>

);

const UserMenu = () => (

<div className="relative">

<button className="flex items-center space-x-2">

<FaUserCircle className="text-2xl text-gray-600" />

<span className="text-gray-700">Admin</span>

</button>

</div>

);

const DashboardView = ({ stats, chartData, chartData2 }) => (

<div className="space-y-6">

{/\* Statistiques \*/}

<div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-4 gap-6">

<StatCard

title="Total Demandes"

value={stats.totalDemandes}

icon={<FaClipboardList />}

color="blue"

/>

<StatCard

title="En Cours"

value={stats.demandesEnCours}

icon={<FaTasks />}

color="yellow"

/>

<StatCard

title="Terminées"

value={stats.demandesTerminees}

icon={<FaChartLine />}

color="green"

/>

<StatCard

title="Utilisateurs"

value={stats.totalUsers}

icon={<FaUsers />}

color="purple"

/>

</div>

{/\* Graphiques \*/}

<div className="grid grid-cols-1 lg:grid-cols-2 gap-6">

<ChartCard title="Statut des Demandes" chart={<Chart type="bar" data={chartData} />} />

<ChartCard title="Répartition par Type" chart={<Chart type="pie" data={chartData2} />} />

</div>

</div>

);

const DemandesView = ({ demandes, handleStatusChange, handleAffecter, selectedRessources, setSelectedRessources }) => (

<div className="bg-white rounded-lg shadow-sm overflow-hidden">

<div className="overflow-x-auto">

<table className="min-w-full divide-y divide-gray-200">

{/\* En-tête du tableau \*/}

<thead className="bg-gray-50">

<tr>

{/\* ... vos en-têtes de colonnes ... \*/}

</tr>

</thead>

{/\* Corps du tableau \*/}

<tbody className="bg-white divide-y divide-gray-200">

{demandes.map((demande, index) => (

<tr key={demande.id} className="hover:bg-gray-50">

{/\* ... vos cellules de données ... \*/}

</tr>

))}

</tbody>

</table>

</div>

</div>

);

const UsersView = ({ users }) => (

<div className="bg-white rounded-lg shadow-sm overflow-hidden">

<table className="min-w-full divide-y divide-gray-200">

<thead className="bg-gray-50">

<tr>

<th className="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">

Nom

</th>

<th className="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">

Email

</th>

<th className="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">

Rôle

</th>

<th className="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">

Statut

</th>

<th className="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">

Actions

</th>

</tr>

</thead>

<tbody className="bg-white divide-y divide-gray-200">

{users.map((user) => (

<tr key={user.id} className="hover:bg-gray-50">

<td className="px-6 py-4 whitespace-nowrap">

<div className="flex items-center">

<div className="h-10 w-10 rounded-full bg-gray-200 flex items-center justify-center">

<FaUserCircle className="text-gray-500 text-xl" />

</div>

<div className="ml-4">

<div className="text-sm font-medium text-gray-900">{user.name}</div>

</div>

</div>

</td>

<td className="px-6 py-4 whitespace-nowrap">

<div className="text-sm text-gray-900">{user.email}</div>

</td>

<td className="px-6 py-4 whitespace-nowrap">

<span className="px-2 inline-flex text-xs leading-5 font-semibold rounded-full bg-green-100 text-green-800">

{user.role}

</span>

</td>

<td className="px-6 py-4 whitespace-nowrap">

<span className="px-2 inline-flex text-xs leading-5 font-semibold rounded-full bg-green-100 text-green-800">

{user.status}

</span>

</td>

<td className="px-6 py-4 whitespace-nowrap text-right text-sm font-medium">

<button className="text-indigo-600 hover:text-indigo-900">Éditer</button>

</td>

</tr>

))}

</tbody>

</table>

</div>

);

const StatCard = ({ title, value, icon, color }) => (

<motion.div

whileHover={{ scale: 1.02 }}

className={`bg-white p-6 rounded-lg shadow-sm border-l-4 border-${color}-500`}

>

<div className="flex items-center justify-between">

<div>

<p className="text-sm text-gray-500">{title}</p>

<p className="text-2xl font-semibold text-gray-800">{value}</p>

</div>

<div className={`text-${color}-500 text-2xl`}>{icon}</div>

</div>

</motion.div>

);

const ChartCard = ({ title, chart }) => (

<div className="bg-white p-6 rounded-lg shadow-sm">

<h3 className="text-lg font-semibold text-gray-800 mb-4">{title}</h3>

{chart}

</div>

);

export default Tables;