

```
import React, { useState, useEffect } from 'react';
import { createClient } from '@supabase/supabase-js';
import { Download, Trash2, Check, RefreshCw, AlertCircle, LogOut, Eye, EyeOff, Co
const supabase = createClient('https://wqtylxrrerhbxagdzftn.supabase.co', 'eyJhbGc

const RESEND_API_KEY = 're_jCpLJKfw_MfWu2jbSzPPgz6pLHQXMAXJb';
const EMAIL_FROM = 'onboarding@resend.dev';
const zones = ['Palm Jumeirah', 'Dubai Marina', 'Downtown', 'Dubai Creek', 'JBR',
const developers = ['Emaar', 'Damac', 'Sobha', 'Meraas', 'Nakheel', 'Dubai Proper
const commissions = [2, 4, 5, 6];
const pipelineStati = ['lead', 'trattativa', 'prenotato', 'venduto', 'incassato'];
const pipelineColors = { lead: 'bg-slate-500', trattativa: 'bg-blue-500', prenotato:
const pipelineLabels = { lead: '🎯 Lead', trattativa: '💬 Trattativa', prenotato:
const clienteStati = ['nuovo', 'contattato', 'interessato', 'trattativa', 'acquisito'];
const clienteStatiColors = { nuovo: 'bg-slate-500', contattato: 'bg-blue-500', interessato: 'bg-yellow-500', trattativa: 'bg-green-500', acquisito: 'bg-red-500' };
const fmt = (n) => (n || 0).toLocaleString('en-AE', { minimumFractionDigits: 0, maximumFractionDigits: 0 });
const fmtDate = (d) => d ? new Date(d).toLocaleDateString('it-IT', { day: '2-digit', month: '2-digit', year: '2-digit' }) : '';
const getWhatsAppLink = (phone, msg = '') => `https://wa.me/${(phone || '').replace(/\D/g, '')}&text=${msg}`;
const getBaseUrl = () => typeof window !== 'undefined' ? window.location.origin : 'http://localhost:3001';

const sendEmail = async (to, subject, html) => { try { await fetch(`https://api.resend.dev/v1/messages`, {
  method: 'post',
  headers: {
    'Content-Type': 'application/json',
    'Authorization': `Bearer ${RESEND_API_KEY}`
  },
  body: JSON.stringify({
    to: to,
    subject: subject,
    content: html
  })
}) } catch (err) { console.error(err) } };

const Logo = ({ size = 'large', centered = false }) => <div className={centered ? 'flex justify-center' : 'flex justify-end'}>
  
</div>

const MobileMenu = ({ isOpen, onClose, tabs, activeTab, setActiveTab, user, onLogout }) => <div style={{ position: 'relative', height: 40 }}>
  <div style={{ position: 'absolute', top: 0, right: 0, width: 40, height: 40, background: 'white', border: '1px solid #ccc', border-radius: 50%, display: flex, align-items: 'center', justify-content: 'center' }}>
    <img alt="Close icon" style={{ width: 15, height: 15 }} />
  </div>
  <div style={{ position: 'absolute', top: 0, left: 0, width: 100%, height: 100%, background: 'white', padding: 10, display: 'grid', grid-template-columns: 1fr 1fr 1fr, grid-template-rows: 1fr 1fr, gap: 10px }}>
    <div style={{ border: '1px solid #ccc', padding: 5px, border-radius: 5px }}>{user}</div>
    <div style={{ border: '1px solid #ccc', padding: 5px, border-radius: 5px }}>{tabs}</div>
    <div style={{ border: '1px solid #ccc', padding: 5px, border-radius: 5px }}>{activeTab}</div>
    <div style={{ border: '1px solid #ccc', padding: 5px, border-radius: 5px }}>{setActiveTab}</div>
    <div style={{ border: '1px solid #ccc', padding: 5px, border-radius: 5px }}>{onLogout}</div>
  </div>
</div>

const SelectWithOther = ({ value, onChange, options, placeholder, className }) => <div style={{ position: 'relative', width: 100% }}>
  <div style={{ position: 'relative', width: 100% }}>
    <input type="text" value={value} style={{ width: 100%, height: 40, padding: 5px, border: '1px solid #ccc', border-radius: 5px }} placeholder={placeholder} />
    <div style={{ position: 'absolute', top: -10px, left: 0, width: 100px, height: 20, background: 'white', border: '1px solid #ccc', border-radius: 5px, display: 'flex', align-items: 'center', justify-content: 'space-between' }}>
      <img alt="More options icon" style={{ width: 15px, height: 15px }} />
      <span>...</span>
    </div>
  </div>
  <div style={{ position: 'absolute', top: 0, left: 0, width: 100%, height: 100%, background: 'white', border: '1px solid #ccc', border-radius: 5px, padding: 10px, display: 'grid', grid-template-columns: 1fr 1fr 1fr, grid-template-rows: 1fr 1fr, gap: 10px }}>
    {options}
  </div>
</div>

const ClientSearch = ({ clienti, onSelect, onCreateNew, selectedClient }) => { const [showForm, setShowForm] = useState(false);
  const [showUserModal, setShowUserModal] = useState(false);
  const [showClienteModal, setShowClienteModal] = useState(false);
  const [editingCliente, setEditingCliente] = useState(null);

  const handleSelect = (clienti) => {
    setSelectedClient(clienti);
    setShowForm(true);
  };

  const handleCreateNew = () => {
    setEditingCliente({ id: null, nome: '', cognome: '' });
    setShowForm(true);
  };

  const handleEdit = (cliente) => {
    setEditingCliente({ ...cliente, id: cliente.id });
    setShowForm(true);
  };

  const handleDelete = (id) => {
    if (confirm(`Are you sure you want to delete ${selectedClient.nome} ${selectedClient.cognome}?`)) {
      supabase
        .from('clients')
        .delete()
        .eq('id', id)
        .then(() => {
          setSelectedClient(null);
          setShowForm(false);
        })
        .catch((err) => console.error(err));
    }
  };

  const handleUpdate = (cliente) => {
    supabase
      .from('clients')
      .update({ nome: cliente.nome, cognome: cliente.cognome })
      .eq('id', cliente.id)
      .then(() => {
        setEditingCliente(null);
        setShowForm(false);
      })
      .catch((err) => console.error(err));
  };

  const handleShowUserModal = (clienti) => {
    setSelectedClient(clienti);
    setShowUserModal(true);
  };

  const handleShowClienteModal = (clienti) => {
    setSelectedClient(clienti);
    setShowClienteModal(true);
  };

  const handleEditClienti = (clienti) => {
    setEditingCliente(clienti);
    setShowForm(true);
  };

  const handleDeleteClienti = (id) => {
    if (confirm(`Are you sure you want to delete ${selectedClient.nome} ${selectedClient.cognome}?`)) {
      supabase
        .from('clients')
        .delete()
        .eq('id', id)
        .then(() => {
          setSelectedClient(null);
          setShowForm(false);
        })
        .catch((err) => console.error(err));
    }
  };

  const handleUpdateClienti = (clienti) => {
    supabase
      .from('clients')
      .update({ nome: clienti.nome, cognome: clienti.cognome })
      .eq('id', clienti.id)
      .then(() => {
        setEditingCliente(null);
        setShowForm(false);
      })
      .catch((err) => console.error(err));
  };

  const handleShowForm = () => {
    setShowForm(true);
  };

  const handleCancelForm = () => {
    setShowForm(false);
  };

  const handleSaveForm = (cliente) => {
    if (cliente.id) {
      handleUpdate(cliente);
    } else {
      handleCreateNew();
    }
  };

  const handleCancelUserModal = () => {
    setShowUserModal(false);
  };

  const handleSaveUserModal = (user) => {
    supabase
      .from('users')
      .insert([user])
      .then(() => {
        setShowUserModal(false);
        handleSelect(user);
      })
      .catch((err) => console.error(err));
  };

  const handleCancelClienteModal = () => {
    setShowClienteModal(false);
  };

  const handleSaveClienteModal = (cliente) => {
    supabase
      .from('clients')
      .insert([cliente])
      .then(() => {
        setShowClienteModal(false);
        handleSelect(cliente);
      })
      .catch((err) => console.error(err));
  };

  return <div style={{ position: 'relative', width: 100% }}>
    <div style={{ position: 'relative', width: 100% }}>
      <input type="text" value={selectedClient ? selectedClient.nome : ''} style={{ width: 100%, height: 40, padding: 5px, border: '1px solid #ccc', border-radius: 5px }} placeholder="Search Clienti" />
      <div style={{ position: 'absolute', top: -10px, left: 0, width: 100px, height: 20, background: 'white', border: '1px solid #ccc', border-radius: 5px, display: 'flex', align-items: 'center', justify-content: 'space-between' }}>
        <img alt="More options icon" style={{ width: 15px, height: 15px }} />
        <span>...</span>
      </div>
    </div>
    <div style={{ position: 'absolute', top: 0, left: 0, width: 100%, height: 100%, background: 'white', border: '1px solid #ccc', border-radius: 5px, padding: 10px, display: 'grid', grid-template-columns: 1fr 1fr 1fr, grid-template-rows: 1fr 1fr, gap: 10px }}>
      {clienti.map((clienti) => <div style={{ border: '1px solid #ccc', padding: 5px, border-radius: 5px }}>{clienti}</div>)}
    </div>
  </div>

```

```

const [editingUser, setEditingUser] = useState(null);
const [adminTab, setAdminTab] = useState('dashboard');
const [filters, setFilters] = useState({ search: '', stato: '' });
const [convertingSale, setConvertingSale] = useState(null);
const [agentTab, setAgentTab] = useState('lista');
const [mobileMenuOpen, setMobileMenuOpen] = useState(false);
const [selectedSales, setSelectedSales] = useState([]);
const [selectedUsers, setSelectedUsers] = useState([]);
const [selectedClienti, setSelectedClienti] = useState([]);
const [showPasswordModal, setShowPasswordModal] = useState(false);
const [selectedCliente, setSelectedCliente] = useState(null);
const [clienteFilters, setClienteFilters] = useState({ search: '', stato: '' });
const [showFilters, setShowFilters] = useState(false);

useEffect(() => { const s = localStorage.getItem('keyprime_user'); if (s) { con

const loadSales = async () => { setLoading(true); const { data } = await supaba
const loadUsers = async () => { const { data } = await supabase.from('user_cred
const loadClienti = async () => { const { data } = await supabase.from('clienti

useEffect(() => { if (user) { loadSales(); loadClienti(); if (user.ruolo === 'a

const handleLogin = async (username, password) => { setLoading(true); setError(
const handleLogout = () => { setUser(null); localStorage.removeItem('keyprime_u
const changePassword = async (np) => { await supabase.from('user_credentials').u

const addLead = async (ld) => { setSaveStatus('Salvataggio...'); let cliente_id

const addSale = async (sd) => { setSaveStatus('Salvataggio...'); let cliente_id

const convertLeadToSale = async (id, v) => { setSaveStatus('Conversione...'); c

const updateSale = async (id, u) => { const s = sales.find(x => x.id === id); i
const deleteSale = async (id) => { if (!window.confirm('Eliminare?')) return; a
const deleteSelectedSales = async () => { if (!selectedSales.length || !window.
const toggleSelectSale = (id) => setSelectedSales(p => p.includes(id) ? p.filte
const selectAllSales = () => setSelectedSales(selectedSales.length === filtered

const createUser = async (d) => { const { error: e } = await supabase.from('use
const updateUser = async (id, d) => { await supabase.from('user_credentials').u
const deleteUser = async (id) => { if (!window.confirm('Eliminare?')) return; a
const deleteSelectedUsers = async () => { if (!selectedUsers.length) return; fo
const toggleSelectUser = (id) => setSelectedUsers(p => p.includes(id) ? p.filte

const createCliente = async (d) => { await supabase.from('clienti').insert([{
const updateCliente = async (id, d) => { await supabase.from('clienti').update(
const deleteCliente = async (id) => { if (!window.confirm('Eliminare?')) return

```



```
function LoginForm({ onLogin, loading }) { const [u, setU] = useState(''); const

function PasswordModal({ currentPassword, onSave, onClose }) { const [op, setOp]

function ConvertModal({ sale, onConvert, onCancel }) { const [v, setV] = useState

function LeadFormClean({ type, userName, clienti, onSubmit }) { const [selectedCl

function SaleFormClean({ type, userName, clienti, onSubmit }) { const [selectedCl

function DashboardTab({ totals, sales, vendite, byStato, byMonth, byAgente, byZon

function VenditeTab({ sales, loading, filters, setFilters, updateSale, deleteSale

function PipelineTab({ byStato, fmt, pipelineColors, pipelineLabels }) { return <

function ClientiTab({ clienti, loadClienti, createCliente, updateCliente, deleteC

function ClienteDetail({ cliente, sales, onBack, onEdit, onDelete, updateCliente,

function ClienteModal({ cliente, onClose, onSave }) { const [f, setF] = useState(

function UserManagement({ users, loadUsers, createUser, updateUser, deleteUser, s

function UserModal({ user, onClose, onSave }) { const [f, setF] = useState(user |
```