

```

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', prenota
const pipelineLabels = { lead: '🎯 Lead', trattativa: '💬 Trattativa', prenotato:
const clienteStati = ['nuovo', 'contattato', 'interessato', 'trattativa', 'acquis
const clienteStatiColors = { nuovo: 'bg-slate-500', contattato: 'bg-blue-500', in
const fmt = (n) => (n || 0).toLocaleString('en-AE', { minimumFractionDigits: 0, m
const fmtDate = (d) => d ? new Date(d).toLocaleDateString('it-IT', { day: '2-digi
const getWhatsAppLink = (phone, msg = '') => `https://wa.me/${(phone || '')}.repla
const getBaseUrl = () => typeof window !== 'undefined' ? window.location.origin :

const sendEmail = async (to, subject, html) => { try { await fetch('https://api.r

const Logo = ({ size = 'large', centered = false }) => <div className={centered ?

const MobileMenu = ({ isOpen, onClose, tabs, activeTab, setActiveTab, user, onLog

const SelectWithOther = ({ value, onChange, options, placeholder, className }) =>

const ClientSearch = ({ clienti, onSelect, onCreateNew, selectedClient }) => { co

export default function App() {
  const [view, setView] = useState('login');
  const [user, setUser] = useState(null);
  const [sales, setSales] = useState([]);
  const [users, setUsers] = useState([]);
  const [clienti, setClienti] = useState([]);
  const [loading, setLoading] = useState(false);
  const [error, setError] = useState(null);
  const [showForm, setShowForm] = useState(null);
  const [saveStatus, setSaveStatus] = useState('');
  const [showUserModal, setShowUserModal] = useState(false);
  const [showClienteModal, setShowClienteModal] = useState(false);
  const [editingCliente, setEditingCliente] = useState(null);

```

```

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: '', b
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').

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

```

```

const deleteSelectedClienti = async () => { if (!selectedClienti.length) return
const toggleSelectCliente = (id) => setSelectedClienti(p => p.includes(id) ? p.

const exportToCSV = () => { let csv = '\u0009Data,Progetto,Cliente,Agente,Valor
const exportClientiToCSV = () => { let csv = '\u0009Nome,Cognome,Telefono,Email

const filteredSales = sales.filter(s => { if (filters.search && !`${s.progetto}
const filteredClienti = clienti.filter(c => { if (clienteFilters.search && !`${
const getClientSales = (clienteId) => sales.filter(s => s.cliente_id === client
const uniqueAgenti = [...new Set(clienti.map(c => c.agente_riferimento).filter(
const ErrorBanner = () => error && <div className="bg-red-500/20 border border-

// LOGIN
if (view === 'login') return <div className="min-h-screen bg-gradient-to-br fro

// AGENTE / SEGNALATORE
if (view === 'agente' || view === 'segnalatore') {
  const type = view;
  const mySales = sales.filter(s => type === 'agente' ? s.agente === user?.nome
  const rate = type === 'agente' ? 0.7 : 0.3;
  const myVendite = mySales.filter(s => s.stato === 'venduto' || s.stato === 'i
  const totalComm = myVendite.reduce((sum, s) => sum + (Number(s.valore) * (s.c
  const pagate = myVendite.filter(s => s.pagato).reduce((sum, s) => sum + (Numb
  const byStato = pipelineStati.reduce((acc, st) => { acc[st] = mySales.filter(
  const agentTabs = [{ id: 'lista', icon: '📋', label: 'Lista' }, { id: 'pipeli

  if (showForm) return <div className="min-h-screen bg-gradient-to-br from-slat

  return <div className="min-h-screen bg-gradient-to-br from-slate-900 via-slat
}

// ADMIN
if (view === 'admin') {
  const vendite = sales.filter(s => s.stato === 'venduto' || s.stato === 'incas
  const totals = sales.reduce((a, s) => { const c = Number(s.valore) * (s.commi
  const byStato = pipelineStati.reduce((a, st) => { a[st] = sales.filter(s => (
  const byMonth = sales.reduce((a, s) => { const m = s.data?.substring(0, 7) ||
  const byAgente = sales.reduce((a, s) => { if (s.agente) a[s.agente] = (a[s.ag
  const byZona = sales.reduce((a, s) => { if (s.zona) a[s.zona] = (a[s.zona] ||
  const adminTabs = [{ id: 'dashboard', icon: '📊', label: 'Dashboard' }, { id:

  return <div className="min-h-screen bg-gradient-to-br from-slate-900 via-slat
}
return null;
}

// COMPONENTS

```

```
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 |
```