

ADMIN DASHBOARD:

```
import React from 'react';

import { Booking } from '../types';

import { LineChart, Line, XAxis, YAxis, CartesianGrid, Tooltip, ResponsiveContainer,
BarChart, Bar } from 'recharts';

import { DollarSign, ShoppingBag, Users as UsersIcon, TrendingUp } from 'lucide-react';

interface AdminDashboardProps {
  bookings: Booking[];
}

const AdminDashboard: React.FC<AdminDashboardProps> = ({ bookings }) => {
  const revenueData = [
    { name: 'Jan', revenue: 4000 },
    { name: 'Feb', revenue: 3000 },
    { name: 'Mar', revenue: 2000 },
    { name: 'Apr', revenue: 2780 },
    { name: 'May', revenue: 1890 },
    { name: 'Jun', revenue: 2390 },
    { name: 'Jul', revenue: 3490 }
  ];

  const categoryData = [
    { name: 'SUVs', bookings: 12 },
    { name: 'Sedans', bookings: 19 },
    { name: 'EVs', bookings: 8 },
    { name: 'Buses', bookings: 3 }
  ];

  const totalRevenue = bookings.reduce((sum, b) => sum + b.totalPrice, 25000);

  return (
    <div className="space-y-6 animate-fade-in">
      <div className="flex flex-col md:flex-row justify-between items-end mb-4">
        <div>
```

`<h2 className="text-2xl font-bold text-gray-900 dark:text-white">Dashboard Overview</h2>`

`<p className="text-gray-500 dark:text-gray-400">Welcome back, Admin. Here's what's happening today.</p>`

`</div>`

`<div className="text-sm text-gray-500 bg-white dark:bg-slate-800 px-3 py-1 rounded-full shadow-sm border border-gray-200 dark:border-slate-700">`

`Last updated: Just now`

`</div>`

`</div>`

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

`<div className="bg-white dark:bg-slate-800 p-6 rounded-xl shadow-sm border border-gray-100 dark:border-slate-700">`

`<div className="flex justify-between items-center mb-4">`

`<div className="p-3 bg-green-100 dark:bg-green-900/30 rounded-lg text-green-600 dark:text-green-400">`

`<DollarSign className="w-6 h-6" />`

`</div>`

`+12%`

`</div>`

`<h3 className="text-3xl font-bold text-gray-900 dark:text-white">${totalRevenue.toLocaleString()}</h3>`

`<p className="text-sm text-gray-500">Total Revenue</p>`

`</div>`

`<div className="bg-white dark:bg-slate-800 p-6 rounded-xl shadow-sm border border-gray-100 dark:border-slate-700">`

`<div className="flex justify-between items-center mb-4">`

`<div className="p-3 bg-blue-100 dark:bg-blue-900/30 rounded-lg text-blue-600 dark:text-blue-400">`

`<ShoppingBag className="w-6 h-6" />`

`</div>`

`+5%`

```
</div>

<h3 className="text-3xl font-bold text-gray-900 dark:text-white">{bookings.length +
142}</h3>

<p className="text-sm text-gray-500">Total Bookings</p>
</div>

<div className="bg-white dark:bg-slate-800 p-6 rounded-xl shadow-sm border border-
gray-100 dark:border-slate-700">

  <div className="flex justify-between items-center mb-4">

    <div className="p-3 bg-purple-100 dark:bg-purple-900/30 rounded-lg text-purple-
600 dark:text-purple-400">

      <UsersIcon className="w-6 h-6" />

    </div>

    <span className="text-xs font-medium text-purple-600 bg-purple-50 px-2 py-1
rounded-full">+18%</span>

  </div>

  <h3 className="text-3xl font-bold text-gray-900 dark:text-white">1,204</h3>

  <p className="text-sm text-gray-500">Active Users</p>
</div>

<div className="bg-white dark:bg-slate-800 p-6 rounded-xl shadow-sm border border-
gray-100 dark:border-slate-700">

  <div className="flex justify-between items-center mb-4">

    <div className="p-3 bg-amber-100 dark:bg-amber-900/30 rounded-lg text-amber-
600 dark:text-amber-400">

      <TrendingUp className="w-6 h-6" />

    </div>

    <span className="text-xs font-medium text-amber-600 bg-amber-50 px-2 py-1
rounded-full">+2%</span>

  </div>

  <h3 className="text-3xl font-bold text-gray-900 dark:text-white">4.8/5</h3>

  <p className="text-sm text-gray-500">Average Rating</p>
</div>
</div>

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

```
<div className="bg-white dark:bg-slate-800 p-6 rounded-xl shadow-sm border border-gray-100 dark:border-slate-700">
```

```
<h3 className="text-lg font-bold text-gray-900 dark:text-white mb-6">Revenue Analytics</h3>
```

```
<div className="h-80 w-full min-h-[320px]">
```

```
<ResponsiveContainer width="100%" height="100%">
```

```
<LineChart data={revenueData}>
```

```
<CartesianGrid strokeDasharray="3 3" stroke="#e5e7eb" opacity={0.3} />
```

```
<XAxis dataKey="name" stroke="#9ca3af" fontSize={12} tickLine={false} axisLine={false} />
```

```
<YAxis stroke="#9ca3af" fontSize={12} tickLine={false} axisLine={false} tickFormatter={(value) => `$$${value}`} />
```

```
<Tooltip contentStyle={{ backgroundColor: '#1e293b', border: 'none', borderRadius: '8px', color: '#fff' }} itemStyle={{ color: '#fff' }} />
```

```
<Line type="monotone" dataKey="revenue" stroke="#4f46e5" strokeWidth={3} dot={{ r: 4, fill: '#4f46e5' }} activeDot={{ r: 8 }} />
```

```
</LineChart>
```

```
</ResponsiveContainer>
```

```
</div>
```

```
</div>
```

```
<div className="bg-white dark:bg-slate-800 p-6 rounded-xl shadow-sm border border-gray-100 dark:border-slate-700">
```

```
<h3 className="text-lg font-bold text-gray-900 dark:text-white mb-6">Vehicle Category Performance</h3>
```

```
<div className="h-80 w-full min-h-[320px]">
```

```
<ResponsiveContainer width="100%" height="100%">
```

```
<BarChart data={categoryData}>
```

```
<CartesianGrid strokeDasharray="3 3" stroke="#e5e7eb" vertical={false} opacity={0.3} />
```

```
<XAxis dataKey="name" stroke="#9ca3af" fontSize={12} tickLine={false} axisLine={false} />
```

```
<YAxis stroke="#9ca3af" fontSize={12} tickLine={false} axisLine={false} />
```

```
<Tooltip cursor={{ fill: 'transparent' }} contentStyle={{ backgroundColor: '#1e293b', border: 'none', borderRadius: '8px', color: '#fff' }} />
```

```

        <Bar dataKey="bookings" fill="#3b82f6" radius={[4, 4, 0, 0]} barSize={40} />
      </BarChart>
    </ResponsiveContainer>
  </div>
</div>
</div>
<div className="bg-white dark:bg-slate-800 rounded-xl shadow-sm border border-gray-100 dark:border-slate-700 overflow-hidden">
  <div className="px-6 py-4 border-b border-gray-100 dark:border-slate-700">
    <h3 className="text-lg font-bold text-gray-900 dark:text-white">Recent Transactions</h3>
  </div>
  <div className="overflow-x-auto">
    <table className="w-full text-left text-sm">
      <thead className="bg-gray-50 dark:bg-slate-900/50 text-gray-500 dark:text-gray-400">
        <tr>
          <th className="px-6 py-3 font-medium">Customer</th>
          <th className="px-6 py-3 font-medium">Item</th>
          <th className="px-6 py-3 font-medium">Type</th>
          <th className="px-6 py-3 font-medium">Date</th>
          <th className="px-6 py-3 font-medium">Amount</th>
          <th className="px-6 py-3 font-medium">Status</th>
        </tr>
      </thead>
      <tbody className="divide-y divide-gray-100 dark:divide-slate-700">
        {bookings.length === 0 ? (
          <tr>
            <td colspan={6} className="px-6 py-4 text-center text-gray-500">No bookings yet.</td>
          </tr>
        ) : (

```

```

    bookings.map((booking) => (
      <tr key={booking.id} className="hover:bg-gray-50 dark: hover:bg-slate-800/50
transition-colors">
        <td className="px-6 py-4 text-gray-900 dark:text-white font-
medium">{booking.customerName}</td>
        <td className="px-6 py-4 text-gray-600 dark:text-gray-
300">{booking.itemName}</td>
        <td className="px-6 py-4">
          <span className={`inline-flex items-center px-2.5 py-0.5 rounded-full text-xs
font-medium ${booking.itemType === 'VEHICLE' ? 'bg-blue-100 text-blue-800 dark:bg-
blue-900 dark:text-blue-300' : 'bg-purple-100 text-purple-800 dark:bg-purple-900 dark:text-
purple-300'}`}>
            {booking.itemType}
          </span>
        </td>
        <td className="px-6 py-4 text-gray-500 dark:text-gray-
400">{booking.date}</td>
        <td className="px-6 py-4 text-gray-900 dark:text-white font-
medium">${booking.totalPrice}</td>
        <td className="px-6 py-4">
          <span className={`inline-flex items-center px-2.5 py-0.5 rounded-full text-xs
font-medium ${booking.status === 'Cancelled' ? 'bg-red-100 text-red-800 dark:bg-red-900
dark:text-red-300' : 'bg-green-100 text-green-800 dark:bg-green-900 dark:text-green-300'}`}>
            {booking.status}
          </span>
        </td>
      </tr>
    ))
  )}
</tbody>
</table>
</div>
</div>
</div>

```

```
);  
};  
  
export default AdminDashboard;
```

AUTH PAGE:

```
import React, { useState } from 'react';  
import { User, UserRole } from '../types';  
  
import { Mail, Lock, User as UserIcon, ArrowRight, CheckCircle, AlertCircle, Eye, EyeOff, ShieldCheck } from 'lucide-react';  
  
interface AuthProps {  
  onLogin: (user: User) => void;  
  onRegister: (user: User, password: string) => boolean;  
  onAuthenticate: (email: string, password: string) => User | null;  
}  
  
const Auth: React.FC<AuthProps> = ({ onLogin, onRegister, onAuthenticate }) => {  
  const [isRegister, setIsRegister] = useState(true);  
  const [email, setEmail] = useState("");  
  const [password, setPassword] = useState("");  
  const [name, setName] = useState("");  
  const [role, setRole] = useState<UserRole>(UserRole.USER);  
  const [showPassword, setShowPassword] = useState(false);  
  const [notification, setNotification] = useState<{ type: 'success' | 'error', message: string } | null>(null);  
  
  const handleSubmit = (e: React.FormEvent) => {  
    e.preventDefault();  
    setNotification(null);  
  
    if (isRegister) {  
      const newUser: User = {  
        id: `u-${Date.now()}`,  
        name: name || (email.split('@')[0]),  
        email: email,
```

```

    role: role,

    avatar: 'https://images.unsplash.com/photo-1535713875002-
d1d0cf377fde?auto=format&fit=crop&w=150&q=80'

  };

  const success = onRegister(newUser, password);
  if (success) {
    setIsRegister(false);
    setNotification({ type: 'success', message: 'Account created successfully! Please log in.'
});
    setPassword(""); // Clear password for security
  } else {
    setNotification({ type: 'error', message: 'User already exists. Please login.' });
  }
} else {
  const user = onAuthenticate(email, password);
  if (user) {
    onLogin(user);
  } else {
    setNotification({ type: 'error', message: 'Invalid email or password. Please try again.' });
  }
}
};

const toggleMode = () => {
  setIsRegister(!isRegister);
  setNotification(null);
  setEmail("");
  setPassword("");
  setName("");
};

return (

```



```

<div className="min-h-screen flex items-center justify-center bg-gradient-to-br from-indigo-900 via-blue-900 to-slate-900 px-4 relative overflow-hidden">

  { /* Background Decorative Elements */ }

  <div className="absolute top-0 left-0 w-full h-full overflow-hidden pointer-events-none">

    <div className="absolute top-10 left-10 w-72 h-72 bg-purple-500 rounded-full mix-blend-multiply filter blur-3xl opacity-20 animate-pulse"></div>

    <div className="absolute bottom-10 right-10 w-72 h-72 bg-indigo-500 rounded-full mix-blend-multiply filter blur-3xl opacity-20 animate-pulse" style={{ animationDelay: '1s' }}></div>

  </div>

  <div className="max-w-4xl w-full bg-white dark:bg-slate-800 rounded-3xl shadow-2xl overflow-hidden flex flex-col md:flex-row relative z-10 animate-fade-in border border-white/10">

    { /* Left Side - Visual / Info */ }

    <div className="md:w-1/2 bg-gradient-to-br from-indigo-600 to-blue-700 p-10 flex flex-col justify-between text-white relative overflow-hidden">

      <div className="absolute inset-0 bg-[url('https://www.transparenttextures.com/patterns/cubes.png')] opacity-10"></div>

      <div className="relative z-10">

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

          <ShieldCheck className="w-8 h-8 text-indigo-200" />

          <span className="text-xl font-bold tracking-wide">TravelPulse</span>

        </div>

        <h2 className="text-4xl font-extrabold mb-4 leading-tight">

          { isRegister ? "Join the Premium Travel Experience." : "Welcome Back!" }

        </h2>

        <p className="text-indigo-100 text-lg opacity-90">

          { isRegister

            ? "Pre-book exclusive vehicles and manage your travel plans with our unified secure platform."

            : "Access your dashboard, manage bookings, and explore new fleet additions." }

        </p>

      </div>

    </div>

  </div>

```

```

</div>

<div className="relative z-10 mt-8 md:mt-0">

  <div className="flex -space-x-3 mb-3">

    

    

    

    <div className="w-10 h-10 rounded-full border-2 border-indigo-500 bg-white
text-indigo-600 flex items-center justify-center text-xs font-bold">+2k</div>

  </div>

  <p className="text-sm text-indigo-200">Trusted by over 2,000 travelers.</p>

</div>

</div>

{ /* Right Side - Form */ }

<div className="md:w-1/2 p-8 md:p-12 bg-white dark:bg-slate-900 flex flex-col
justify-center">

  <div className="mb-6">

    <h3 className="text-2xl font-bold text-gray-900 dark:text-white mb-1">{isRegister
? "Create Account" : "Secure Login"}</h3>

    <p className="text-gray-500 dark:text-gray-400 text-sm">Please enter your details
to continue.</p>

  </div>

  {notification && (

    <div className={`mb-6 p-4 rounded-xl flex items-start animate-fade-in
${notification.type === 'success' ? 'bg-green-50 text-green-700 border border-green-200' :
'bg-red-50 text-red-700 border border-red-200'}`}>

      {notification.type === 'success' ? <CheckCircle className="w-5 h-5 mr-3 mt-0.5
flex-shrink-0" /> : <AlertCircle className="w-5 h-5 mr-3 mt-0.5 flex-shrink-0" />}

      <span className="text-sm font-medium">{notification.message}</span>

    </div>
  )}

```

```

    })

    <form onSubmit={handleSubmit} className="space-y-5">
      {isRegister && (
        <div className="space-y-1">
          <label className="text-sm font-semibold text-gray-700 dark:text-gray-300 ml-1">Full Name</label>

          <div className="relative group">
            <UserIcon className="absolute left-3 top-3.5 w-5 h-5 text-gray-400 group-focus-within:text-indigo-500 transition-colors" />

            <input
              type="text"
              value={name}
              onChange={(e) => setName(e.target.value)}
              required
              className="w-full pl-10 pr-4 py-3 rounded-xl border border-gray-200 dark:border-slate-700 bg-gray-50 dark:bg-slate-800 text-gray-900 dark:text-white focus:ring-2 focus:ring-indigo-500 focus:border-transparent outline-none transition-all"
              placeholder="John Doe"
            />
          </div>
        </div>
      )}

      <div className="space-y-1">
        <label className="text-sm font-semibold text-gray-700 dark:text-gray-300 ml-1">Email Address</label>

        <div className="relative group">
          <Mail className="absolute left-3 top-3.5 w-5 h-5 text-gray-400 group-focus-within:text-indigo-500 transition-colors" />

          <input
            type="email"
            value={email}
            onChange={(e) => setEmail(e.target.value)}

```

```

        required

        className="w-full pl-10 pr-4 py-3 rounded-xl border border-gray-200
dark:border-slate-700 bg-gray-50 dark:bg-slate-800 text-gray-900 dark:text-white focus:ring-
2 focus:ring-indigo-500 focus:border-transparent outline-none transition-all"

        placeholder="name@example.com"

    />
</div>
</div>

<div className="space-y-1">

    <label className="text-sm font-semibold text-gray-700 dark:text-gray-300 ml-
1">Password</label>

    <div className="relative group">

        <Lock className="absolute left-3 top-3.5 w-5 h-5 text-gray-400 group-focus-
within:text-indigo-500 transition-colors" />

        <input

            type={showPassword ? "text" : "password"}

            value={password}

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

            required

            className="w-full pl-10 pr-12 py-3 rounded-xl border border-gray-200
dark:border-slate-700 bg-gray-50 dark:bg-slate-800 text-gray-900 dark:text-white focus:ring-
2 focus:ring-indigo-500 focus:border-transparent outline-none transition-all"

            placeholder="••••••••"

        />

        <button

            type="button"

            onClick={() => setShowPassword(!showPassword)}

            className="absolute right-3 top-3.5 text-gray-400 hover:text-gray-600
dark:hover:text-gray-200 focus:outline-none"

        >

            {showPassword ? <EyeOff className="w-5 h-5" /> : <Eye className="w-5 h-
5" />}

```

```

        </button>

    </div>

</div>

{isRegister && (
    <div className="pt-2">
        <span className="text-xs font-semibold text-gray-500 dark:text-gray-400 uppercase tracking-wider mb-2 block">Account Type</span>

        <div className="flex gap-4">
            <label className={`flex-1 cursor-pointer border rounded-xl p-3 flex items-center justify-center transition-all ${role === UserRole.USER ? 'border-indigo-500 bg-indigo-50 dark:bg-indigo-900/20 text-indigo-700 dark:text-indigo-300' : 'border-gray-200 dark:border-slate-700 text-gray-600 dark:text-gray-400'}`}>
                <input
                    type="radio"
                    name="role"
                    className="hidden"
                    checked={role === UserRole.USER}
                    onChange={() => setRole(UserRole.USER)}
                />

                <span className="text-sm font-medium">Customer</span>
            </label>

            <label className={`flex-1 cursor-pointer border rounded-xl p-3 flex items-center justify-center transition-all ${role === UserRole.ADMIN ? 'border-indigo-500 bg-indigo-50 dark:bg-indigo-900/20 text-indigo-700 dark:text-indigo-300' : 'border-gray-200 dark:border-slate-700 text-gray-600 dark:text-gray-400'}`}>
                <input
                    type="radio"
                    name="role"
                    className="hidden"
                    checked={role === UserRole.ADMIN}
                    onChange={() => setRole(UserRole.ADMIN)}
                />

                <span className="text-sm font-medium">Admin</span>
            </label>
        </div>
    </div>
)

```

```

        </label>

      </div>

    </div>

  )}

  <button
    type="submit"

    className="w-full bg-indigo-600 hover:bg-indigo-700 text-white font-bold py-3.5
rounded-xl transition-all flex items-center justify-center mt-4 shadow-lg shadow-indigo-
500/30 active:scale-[0.98]"

    >

      {isRegister ? 'Sign Up' : 'Sign In'} <ArrowRight className="w-4 h-4 ml-2" />

    </button>

  </form>

  <div className="mt-8 text-center pt-6 border-t border-gray-100 dark:border-slate-
800">

    <p className="text-sm text-gray-500 dark:text-gray-400">

      {isRegister ? 'Already have an account?' : "Don't have an account?" }{ ' '}

      <button

        onClick={toggleMode}

        className="text-indigo-600 font-bold hover:text-indigo-700 transition-colors"

        >

          {isRegister ? 'Log In' : 'Register Now'}

        </button>

      </p>

    </div>

  </div>

</div>

);

};

export default Auth;

```

HOME PAGE:

```
import React from 'react';

import { MapPin, Calendar, Search, Car, CheckCircle } from 'lucide-react';

import { MOCK_VEHICLES } from '../constants';

import { VehicleCategory, Vehicle } from '../types';

import VehicleCard from './VehicleCard';

interface HomeProps {

  user: any;

  setView: (view: string) => void;

  onBook: (vehicle: Vehicle, type: 'VEHICLE') => void;

}

const Home: React.FC<HomeProps> = ({ user, setView, onBook }) => {

  return (

    <div className="animate-fade-in">

      <div className="relative bg-slate-900 h-[600px] flex items-center overflow-hidden">

        <div className="absolute inset-0">

          

          <div className="absolute inset-0 bg-gradient-to-t from-slate-900 via-transparent to-slate-900/50" />

        </div>

        <div className="relative max-w-7xl mx-auto px-4 sm:px-6 lg:px-8 flex flex-col items-center text-center">

          <h1 className="text-4xl md:text-6xl font-extrabold text-white tracking-tight mb-6">

            Vehicle Pre-Booking <span className="text-indigo-400">System</span>

          </h1>

        </div>

      </div>

    </div>

  );

};
```

```
<p className="text-xl text-gray-300 max-w-2xl mb-10">
```

Your one-stop destination to rent vehicles, pre-book showroom cars, and explore curated tour packages.

```
</p>
```

```
<div className="bg-white p-2 rounded-xl shadow-2xl flex flex-col md:flex-row gap-2 max-w-3xl w-full">
```

```
<div className="flex-1 relative">
```

```
<MapPin className="absolute left-3 top-3.5 text-gray-400 w-5 h-5" />
```

```
<input type="text" placeholder="Where to?" className="w-full pl-10 pr-4 py-3 bg-gray-50 rounded-lg outline-none focus:ring-2 focus:ring-indigo-500 text-gray-900 placeholder-gray-500" />
```

```
</div>
```

```
<div className="flex-1 relative">
```

```
<Calendar className="absolute left-3 top-3.5 text-gray-400 w-5 h-5" />
```

```
<input type="date" className="w-full pl-10 pr-4 py-3 bg-gray-50 rounded-lg outline-none focus:ring-2 focus:ring-indigo-500 text-gray-900" />
```

```
</div>
```

```
<button
```

```
  onClick={() => setView('vehicles')}
```

```
  className="bg-indigo-600 hover:bg-indigo-700 text-white font-bold py-3 px-8 rounded-lg transition-colors flex items-center justify-center"
```

```
>
```

```
<Search className="w-5 h-5 mr-2" /> Search
```

```
</button>
```

```
</div>
```

```
</div>
```

```
</div>
```

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

```
<h2 className="text-3xl font-bold text-gray-900 dark:text-white mb-12 text-center">Our Services</h2>
```

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

```
<div onClick={() => setView('vehicles')} className="cursor-pointer group p-6 bg-white dark:bg-slate-800 rounded-2xl shadow-sm hover:shadow-xl transition-all border border-gray-100 dark:border-slate-700">
```



```
<div className="w-16 h-16 mx-auto bg-blue-100 dark:bg-blue-900/30 text-blue-600 rounded-full flex items-center justify-center mb-4 group-hover:scale-110 transition-transform">
```

```
<Car className="w-8 h-8" />
```

```
</div>
```

```
<h3 className="text-xl font-bold mb-2 text-gray-900 dark:text-white">Vehicle Rentals</h3>
```

```
<p className="text-gray-500 dark:text-gray-400">Cars, Bikes, SUVs, and Buses for your daily or outstation needs.</p>
```

```
</div>
```

```
<div onClick={() => setView('showroom')} className="cursor-pointer group p-6 bg-white dark:bg-slate-800 rounded-2xl shadow-sm hover:shadow-xl transition-all border border-gray-100 dark:border-slate-700">
```

```
<div className="w-16 h-16 mx-auto bg-purple-100 dark:bg-purple-900/30 text-purple-600 rounded-full flex items-center justify-center mb-4 group-hover:scale-110 transition-transform">
```

```
<CheckCircle className="w-8 h-8" />
```

```
</div>
```

```
<h3 className="text-xl font-bold mb-2 text-gray-900 dark:text-white">Showroom Pre-Booking</h3>
```

```
<p className="text-gray-500 dark:text-gray-400">Book new vehicles directly from the showroom with a small token amount.</p>
```

```
</div>
```

```
<div onClick={() => setView('packages')} className="cursor-pointer group p-6 bg-white dark:bg-slate-800 rounded-2xl shadow-sm hover:shadow-xl transition-all border border-gray-100 dark:border-slate-700">
```

```
<div className="w-16 h-16 mx-auto bg-green-100 dark:bg-green-900/30 text-green-600 rounded-full flex items-center justify-center mb-4 group-hover:scale-110 transition-transform">
```

```
<MapPin className="w-8 h-8" />
```

```
</div>
```

```
<h3 className="text-xl font-bold mb-2 text-gray-900 dark:text-white">Trips & Tours</h3>
```

```
<p className="text-gray-500 dark:text-gray-400">Curated holiday packages and pilgrimage tours for friends and family.</p>
```

```
</div>
```

```

    </div>

  </div>

  <div className="bg-gray-50 dark:bg-slate-900/50 py-16">

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

      <h2 className="text-3xl font-bold text-gray-900 dark:text-white mb-8">Popular
Rentals</h2>

      <div className="grid grid-cols-1 md:grid-cols-3 gap-6">

        {MOCK_VEHICLES.filter(v => v.category ===
VehicleCategory.RENTAL).slice(0, 3).map(v => (
          <VehicleCard key={v.id} vehicle={v} onBook={(veh) => onBook(veh,
'VEHICLE')} isLoggedIn={!!user} />

          )))}

      </div>

    </div>

  </div>

</div>

);

};

```

export default Home;

LAYOUT PAGE:

```

import React from 'react';

import { Menu, X, Sun, Moon, LogIn, LogOut, Car, LayoutDashboard, Map, Sparkles,
ShoppingBag, List } from 'lucide-react';

import { User, UserRole } from '../types';

interface LayoutProps {
  children: React.ReactNode;
  darkMode: boolean;
  toggleDarkMode: () => void;
  currentUser: User | null;
  onLogin: () => void;
  onLogout: () => void;
  currentView: string;

```

```

    setView: (view: string) => void;
  }
const Layout: React.FC<LayoutProps> = ({
  children, darkMode, toggleDarkMode, currentUser, onLogin, onLogout, currentView,
  setView
}) => {
  const [isMobileMenuOpen, setIsMobileMenuOpen] = React.useState(false);
  const navItems = [
    { id: 'home', label: 'Home', icon: <Map className="w-4 h-4" /> },
    { id: 'vehicles', label: 'Rentals', icon: <Car className="w-4 h-4" /> },
    { id: 'showroom', label: 'Showroom', icon: <ShoppingBag className="w-4 h-4" /> },
    { id: 'packages', label: 'Trips & Tours', icon: <Map className="w-4 h-4" /> },
    { id: 'ai-planner', label: 'AI Planner', icon: <Sparkles className="w-4 h-4 text-purple-500" /> },
  ];
  if (currentUser?.role === UserRole.ADMIN) {
    navItems.push({ id: 'admin', label: 'Dashboard', icon: <LayoutDashboard className="w-4 h-4" /> });
  } else if (currentUser) {
    // For normal users, show My Bookings
    navItems.push({ id: 'my-bookings', label: 'My Bookings', icon: <List className="w-4 h-4" /> });
  }
  return (
    <div className="min-h-screen flex flex-col bg-gray-50 dark:bg-slate-900 transition-colors duration-300">
      { /* Navbar */ }
      <nav className="sticky top-0 z-50 bg-white/80 dark:bg-slate-900/80 backdrop-blur-md border-b border-gray-200 dark:border-slate-700">
        <div className="max-w-7xl mx-auto px-4 sm:px-6 lg:px-8">
          <div className="flex justify-between h-16 items-center">
            { /* Logo */ }

```

```

    <div className="flex items-center cursor-pointer" onClick={() =>
setView('home')}>
      <Car className="h-8 w-8 text-indigo-600 dark:text-indigo-400" />
      <span className="ml-2 text-xl font-bold bg-clip-text text-transparent bg-gradient-
to-r from-indigo-600 to-blue-500">
        TravelPulse
      </span>
    </div>

    { /* Desktop Nav */ }

    <div className="hidden md:flex space-x-8 items-center">
      { navItems.map((item) => (
        <button
          key={item.id}
          onClick={() => setView(item.id)}
          className={`flex items-center space-x-1 px-3 py-2 rounded-md text-sm font-
medium transition-colors ${
            currentView === item.id
              ? 'text-indigo-600 dark:text-indigo-400 bg-indigo-50 dark:bg-slate-800'
              : 'text-gray-700 dark:text-gray-300 hover:text-indigo-600 dark:hover:text-
indigo-400'
            }`}
          >
            {item.icon}
            <span>{item.label}</span>
          </button>
        ) ) }
      <button onClick={toggleDarkMode} className="p-2 rounded-full hover:bg-gray-
100 dark:hover:bg-slate-800 text-gray-600 dark:text-gray-400">
        { darkMode ? <Sun className="w-5 h-5" /> : <Moon className="w-5 h-5" /> }
      </button>
      { currentUser ? (
        <div className="flex items-center space-x-3">

```

```

        <div className="flex flex-col items-end">
            <span className="text-sm font-semibold text-gray-800 dark:text-white">{currentUser.name}</span>
            <span className="text-xs text-gray-500 dark:text-gray-400">{currentUser.role}</span>
        </div>
        <button
            onClick={onLogout}
            className="flex items-center px-4 py-2 text-sm font-medium text-white bg-red-500 hover:bg-red-600 rounded-lg transition shadow-sm"
        >
            <LogOut className="w-4 h-4 mr-1" /> Logout
        </button>
    </div>
) : (
    <button
        onClick={onLogin}
        className="flex items-center px-4 py-2 text-sm font-medium text-white bg-indigo-600 hover:bg-indigo-700 rounded-lg transition shadow-lg shadow-indigo-500/30"
    >
        <LogIn className="w-4 h-4 mr-1" /> Login
    </button>
)}
</div>
{/* Mobile Menu Button */}
<div className="flex items-center md:hidden">
    <button
        onClick={() => setIsMobileMenuOpen(!isMobileMenuOpen)}
        className="p-2 text-gray-600 dark:text-gray-300"
    >
        {isMobileMenuOpen ? <X className="w-6 h-6" /> : <Menu className="w-6 h-6" />}
    </div>

```

```

        </button>

    </div>

</div>

</div>

{ /* Mobile Menu */}

{isMobileMenuOpen && (

    <div className="md:hidden bg-white dark:bg-slate-900 border-t border-gray-200
dark:border-slate-700">

        <div className="px-2 pt-2 pb-3 space-y-1 sm:px-3">

            {navItems.map((item) => (

                <button

                    key={item.id}

                    onClick={() => {

                        setView(item.id);

                        setIsMobileMenuOpen(false);

                    }}

                    className="block w-full text-left px-3 py-2 rounded-md text-base font-medium
text-gray-700 dark:text-gray-300 hover:bg-gray-100 dark:hover:bg-slate-800"

                >

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

                        {item.icon}

                        <span>{item.label}</span>

                    </span>

                </button>

            ))}

            <div className="border-t border-gray-200 dark:border-slate-700 pt-4 mt-2">

                <button onClick={toggleDarkMode} className="w-full flex items-center px-3
py-2 text-gray-700 dark:text-gray-300 hover:bg-gray-100 dark:hover:bg-slate-800">

                    {darkMode ? <Sun className="w-5 h-5 mr-2" /> : <Moon className="w-5 h-5
mr-2" />}

                    <span>{darkMode ? 'Light Mode' : 'Dark Mode'}</span>

                </button>

```

```

      {currentUser ? (
        <button onClick={onLogout} className="w-full flex items-center px-3 py-2
text-red-600 hover:bg-red-50 dark:hover:bg-slate-800 mt-2">
          <LogOut className="w-5 h-5 mr-2" /> Logout
        </button>
      ) : (
        <button onClick={onLogin} className="w-full flex items-center px-3 py-2
text-indigo-600 hover:bg-indigo-50 dark:hover:bg-slate-800 mt-2">
          <LogIn className="w-5 h-5 mr-2" /> Login
        </button>
      )}
    </div>
  </div>
</div>
)}
</nav>
{/* Main Content */}
<main className="flex-grow">
  {children}
</main>
{/* Footer */}
<footer className="bg-white dark:bg-slate-950 border-t border-gray-200 dark:border-
slate-800 py-12">
  <div className="max-w-7xl mx-auto px-4 sm:px-6 lg:px-8">
    <div className="grid grid-cols-1 md:grid-cols-4 gap-8">
      <div className="col-span-1 md:col-span-1">
        <div className="flex items-center mb-4">
          <Car className="h-6 w-6 text-indigo-600" />
          <span className="ml-2 text-lg font-bold text-gray-900 dark:text-
white">TravelPulse</span>
        </div>
        <p className="text-gray-500 dark:text-gray-400 text-sm">

```

Advanced Vehicle Pre-Booking & Travel Management System powered by AI.

</p>

</div>

<div>

<h3 className="text-sm font-semibold text-gray-900 dark:text-white uppercase tracking-wider mb-4">Services</h3>

<ul className="space-y-2 text-gray-500 dark:text-gray-400 text-sm">

Vehicle Rentals

Tour Packages

Showroom Pre-booking

Wedding Fleets

</div>

<div>

<h3 className="text-sm font-semibold text-gray-900 dark:text-white uppercase tracking-wider mb-4">Support</h3>

<ul className="space-y-2 text-gray-500 dark:text-gray-400 text-sm">

Help Center

Terms of Service

Privacy Policy

Contact Us

</div>

<div>

<h3 className="text-sm font-semibold text-gray-900 dark:text-white uppercase tracking-wider mb-4">Contact</h3>

<p className="text-gray-500 dark:text-gray-400 text-sm">

123 Innovation Drive

Tech City, TC 90210

support@travelpulse.com

</p>


```

        </div>

        </div>

        <div className="mt-8 pt-8 border-t border-gray-200 dark:border-slate-800 text-center text-sm text-gray-400">

            &copy; 2024 TravelPulse Inc. All rights reserved.

        </div>

    </div>

    </footer>

</div>

);

};

export default Layout;

```

MY BOOKING PAGE:

```

import React from 'react';

import { Booking, User } from '../types';

import { ShoppingBag, CheckCircle, XCircle, Calendar, Clock } from 'lucide-react';

interface MyBookingsProps {

    user: User | null;

    bookings: Booking[];

    onCancel: (id: string) => void;

    setView: (view: string) => void;

}

const MyBookings: React.FC<MyBookingsProps> = ({ user, bookings, onCancel, setView }) => {

    const userBookings = bookings.filter(b => b.customerName === user?.name);

    return (

        <div className="max-w-4xl mx-auto px-4 py-8 animate-fade-in">

            <h1 className="text-3xl font-bold text-gray-900 dark:text-white mb-6">My Bookings</h1>

            <div className="bg-white dark:bg-slate-800 rounded-xl shadow-sm border border-gray-100 dark:border-slate-700 overflow-hidden">

                {userBookings.length === 0 ? (

```

```

    <div className="p-10 text-center text-gray-500">
      <ShoppingBag className="w-12 h-12 mx-auto mb-3 opacity-20" />
      <p>You haven't made any bookings yet.</p>
      <button onClick={() => setView('vehicles')} className="text-indigo-600
font-medium mt-2 hover:underline">Browse Vehicles</button>
    </div>
  ): (
    <div className="divide-y divide-gray-100 dark:divide-slate-700">
      {userBookings.map(booking => (
        <div key={booking.id} className="p-6 flex flex-col md:flex-row justify-
between items-center hover:bg-gray-50 dark:hover:bg-slate-800/50">
          <div className="flex items-center space-x-4 mb-4 md:mb-0 w-full
md:w-auto">
            <div className={`p-3 rounded-full flex-shrink-0 ${
              booking.status === 'Cancelled' ? 'bg-red-100 text-red-600 dark:bg-
red-900/30' :
              booking.itemType === 'VEHICLE' ? 'bg-blue-100 text-blue-600' :
              booking.itemType === 'SHOWROOM_PREBOOK' ? 'bg-purple-
100 text-purple-600' : 'bg-green-100 text-green-600'
            }`>
              {booking.status === 'Cancelled' ? <XCircle className="w-6 h-6"
/> : <CheckCircle className="w-6 h-6" />}
            </div>
            <div className="flex-grow">
              <h3 className="font-bold text-gray-900 dark:text-white text-
lg">{booking.itemName}</h3>
              <div className="text-sm text-gray-500 space-y-1">
                <p className="flex items-center"><Clock className="w-3 h-3
mr-1" /> Booked on: {booking.date}</p>
                { /* Show Rental Dates if available */ }
                {booking.startDate && booking.endDate && (
                  <p className="flex items-center text-indigo-600 dark:text-
indigo-400 font-medium">
                    <Calendar className="w-3 h-3 mr-1" />

```

```

        Rental: {booking.startDate} to {booking.endDate}
    </p>
    </div>
</div>
</div>
<div className="flex flex-row md:flex-col items-end gap-4 md:gap-2
w-full md:w-auto justify-between md:justify-end">
    <div className="text-left md:text-right">
        <span className="block text-xl font-bold text-gray-900 dark:text-
white">${booking.totalPrice}</span>
        <span className={`inline-block px-2 py-1 text-xs rounded-full
font-medium mt-1 ${
            booking.status === 'Cancelled'
            ? 'bg-red-100 text-red-800 dark:bg-red-900 dark:text-red-300'
            : 'bg-green-100 text-green-800 dark:bg-green-900 dark:text-
green-300'
        }}>
            {booking.status}
        </span>
    </div>
    {booking.status !== 'Cancelled' && (
        <button
            onClick={() => onCancel(booking.id)}
            className="px-4 py-2 text-sm font-medium text-red-600 border
border-red-200 hover:bg-red-50 dark:hover:bg-red-900/20 rounded-lg transition-colors"
        >
            Cancel Booking
        </button>
    )}
</div>
</div>

```

```

        )))}
      </div>

    })
  </div>
</div>

);
};

export default MyBookings;

```

PACKAGES:

```

import React from 'react';

import { MOCK_PACKAGES } from '../constants';

import { TravelPackage, User } from '../types';

import { MapPin } from 'lucide-react';

interface PackagesProps {
  user: User | null;
  onBook: (pkg: TravelPackage, type: 'PACKAGE') => void;
}

const Packages: React.FC<PackagesProps> = ({ user, onBook }) => {
  return (
    <div className="max-w-7xl mx-auto px-4 py-8 animate-fade-in">
      <div className="text-center mb-12">
        <h1 className="text-3xl font-bold text-gray-900 dark:text-white mb-4">Trips &
Tour Packages</h1>
        <p className="text-gray-500 dark:text-gray-400">Explore the world with our
hand-picked holiday experiences.</p>
      </div>
      <div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-8">
        {MOCK_PACKAGES.map(pkg => (
          <div key={pkg.id} className="bg-white dark:bg-slate-800 rounded-xl shadow-
lg overflow-hidden border border-gray-100 dark:border-slate-700 flex flex-col group
hover:shadow-2xl transition-all">
            <div className="relative h-56 overflow-hidden">

```

```

        <img src={pkg.image} alt={pkg.title} className="h-full w-full object-
cover group-hover:scale-105 transition-transform duration-500" />

        <div className="absolute bottom-0 left-0 right-0 bg-gradient-to-t from-
black/80 to-transparent p-4">

            <span className="text-white font-bold flex items-center">

                <MapPin className="w-4 h-4 mr-1 text-indigo-400" />
{pkg.location}

            </span>

        </div>

    </div>

    <div className="p-6 flex flex-col flex-grow">

        <div className="flex justify-between items-start mb-2">

            <h3 className="text-xl font-bold text-gray-900 dark:text-
white">{pkg.title}</h3>

            <span className="bg-blue-100 text-blue-800 text-xs px-2 py-1
rounded-full whitespace-nowrap ml-2">{pkg.duration}</span>

        </div>

        <p className="text-gray-500 dark:text-gray-400 text-sm mb-4 line-clamp-
3">{pkg.description}</p>

        <div className="mb-4">

            <span className="text-xs font-semibold uppercase text-gray-
400">Highlights</span>

            <div className="flex flex-wrap gap-2 mt-2">

                {pkg.highlights.map(h => (

                    <span key={h} className="text-xs bg-gray-100 dark:bg-slate-700
text-gray-600 dark:text-gray-300 px-2 py-1 rounded">{h}</span>

                ))}

            </div>

        </div>

        <div className="mt-auto pt-4 border-t border-gray-100 dark:border-slate-
700 flex justify-between items-center">

            <div>

                <span className="text-2xl font-bold text-gray-900 dark:text-
white">${pkg.price}</span>

```

```

        <span className="text-xs text-gray-500 block">per person</span>
      </div>
      <button
        onClick={() => onBook(pkg, 'PACKAGE')}
        className="bg-indigo-600 hover:bg-indigo-700 text-white px-4 py-2
rounded-lg font-medium transition-colors shadow-lg shadow-indigo-500/20"
      >
        {user ? 'Book Tour' : 'Login to Book'}
      </button>
    </div>
  </div>
</div>
  )))}
</div>
</div>
);
};

```

export default Packages;

RENTALS PAGE:

```

import React, { useState, useEffect } from 'react';
import { MOCK_VEHICLES, getVehicleImageByType } from '../constants';
import { VehicleCategory, VehicleType, Vehicle, User } from '../types';
import VehicleCard from './VehicleCard';
import { getAIRRecommendations } from '../services/geminiService';
import { Loader2, Sparkles, Car } from 'lucide-react';
interface RentalsProps {
  user: User | null;
  onBook: (vehicle: Vehicle, type: 'VEHICLE') => void;
  setView: (view: string) => void;
}

```

```

const Rentals: React.FC<RentalsProps> = ({ user, onBook, setView }) => {
  const [searchTerm, setSearchTerm] = useState("");
  const [filterType, setFilterType] = useState('All');
  const [aiVehicles, setAiVehicles] = useState<Vehicle[]>([]);
  const [isAiLoading, setIsAiLoading] = useState(false);
  const [lastSearchedTerm, setLastSearchedTerm] = useState("");
  const filtered = MOCK_VEHICLES.filter(v =>
    v.category === VehicleCategory.RENTAL &&
    (filterType === 'All' || v.type === filterType) &&
    v.name.toLowerCase().includes(searchTerm.toLowerCase())
  );
  // AI Recommendation Logic moved here to be a proper Hook usage
  useEffect(() => {
    const triggerAI = async () => {
      if (filtered.length === 0 && searchTerm.length > 2 && searchTerm !==
lastSearchedTerm && !isAiLoading) {
        setIsAiLoading(true);
        setLastSearchedTerm(searchTerm);
        const recs = await getAIRecommendations(searchTerm);
        setAiVehicles(recs);
        setIsAiLoading(false);
      } else if (filtered.length > 0) {
        setAiVehicles([]);
      }
    };
    const timer = setTimeout(triggerAI, 1000);
    return () => clearTimeout(timer);
  }, [searchTerm, filtered.length, lastSearchedTerm, isAiLoading]);

  return (
    <div className="max-w-7xl mx-auto px-4 py-8 animate-fade-in">

```

```

4">
    <div>
        <h1 className="text-3xl font-bold text-gray-900 dark:text-white">Vehicle
Rentals</h1>
        <p className="text-gray-500 dark:text-gray-400">Choose from our wide range
of cars, bikes, and buses.</p>
    </div>
    <div className="flex flex-col sm:flex-row gap-4 w-full md:w-auto">
        <select
            className="bg-white dark:bg-slate-800 border border-gray-300 dark:border-
slate-700 rounded-lg px-4 py-2 outline-none"
            value={filterType}
            onChange={(e) => setFilterType(e.target.value)}
        >
            <option value="All">All Types</option>
            <option value={ VehicleType.CAR }>Cars</option>
            <option value={ VehicleType.SUV }>SUVs</option>
            <option value={ VehicleType.BIKE }>Bikes</option>
            <option value={ VehicleType.SCOOTER }>Scooters</option>
            <option value={ VehicleType.BUS }>Buses</option>
            <option value={ VehicleType.MINIBUS }>Tempo Travellers</option>
            <option value={ VehicleType.VAN }>Vans</option>
            <option value={ VehicleType.EV }>Electric</option>
            <option value={ VehicleType.LUXURY }>Luxury</option>
            <option value={ VehicleType.TRUCK }>Trucks</option>
        </select>
        <input
            type="text"
            placeholder="Search vehicles..."
            className="bg-white dark:bg-slate-800 border border-gray-300 dark:border-
slate-700 rounded-lg px-4 py-2 outline-none w-full"

```



```

        value={searchTerm}

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

    />

</div>

</div>

{/* Local Results */}

<div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-8">

    {filtered.map(v => (

        <VehicleCard key={v.id} vehicle={v} onBook={(veh) => onBook(veh,
'VEHICLE')} isLoggedIn={!!user} />

    ))}

</div>

{/* AI Recommendations */}

{filtered.length === 0 && (

    <div className="mt-8">

        {isAiLoading ? (

            <div className="flex flex-col items-center justify-center py-20">

                <Loader2 className="w-10 h-10 animate-spin text-indigo-600 mb-4" />

                <p className="text-gray-500">Checking our global fleet network for
"{searchTerm}"...</p>

            </div>

        ) : aiVehicles.length > 0 ? (

            <div className="animate-fade-in">

                <div className="flex items-center gap-2 mb-6 p-4 bg-indigo-50 dark:bg-
indigo-900/20 rounded-lg border border-indigo-100 dark:border-indigo-800">

                    <Sparkles className="w-5 h-5 text-indigo-600" />

                    <div>

                        <h3 className="font-semibold text-indigo-900 dark:text-indigo-
100">We couldn't find exactly "{searchTerm}", but here are some AI
recommendations:</h3>

                    </div>

                </div>

            </div>

        ) : null}

    </div>

)}


```

```

    <div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-8">
      {aiVehicles.map((v) => (
        <VehicleCard
          key={v.id}
          vehicle={{...v, image: getVehicleImageByType(v.type)}}
          onBook={(veh) => onBook(veh, 'VEHICLE')}
          isLoggedIn={!user}
        />
      ))}
    </div>
  </div>
) : searchTerm.length > 2 && (
  <div className="text-center py-20 text-gray-500">
    <Car className="w-12 h-12 mx-auto mb-3 opacity-20" />
    <p>No vehicles found. Try searching for "Car", "Bike", or "Bus".</p>
  </div>
  )}
</div>
)}
</div>
)
</div>
);
};

```

export default Rentals;

SHOWROOM PAGE:

```

import React from 'react';
import { MOCK_VEHICLES } from '../constants';
import { VehicleCategory, Vehicle, User } from '../types';
import VehicleCard from './VehicleCard';
interface ShowroomProps {
  user: User | null;

```

```

    onBook: (vehicle: Vehicle, type: 'SHOWROOM_PREBOOK') => void;
  }

const Showroom: React.FC<ShowroomProps> = ({ user, onBook }) => {
  const filtered = MOCK_VEHICLES.filter(v => v.category ===
VehicleCategory.SHOWROOM);

  return (
    <div className="max-w-7xl mx-auto px-4 py-8 animate-fade-in">
      <div className="text-center mb-12">
        <span className="bg-purple-100 text-purple-800 text-sm font-bold px-3 py-1
rounded-full uppercase tracking-wider mb-2 inline-block">Brand New</span>
        <h1 className="text-4xl font-extrabold text-gray-900 dark:text-white mb-
4">Showroom Pre-Booking</h1>
        <p className="text-lg text-gray-600 dark:text-gray-300 max-w-2xl mx-auto">
          Secure the latest models with a small token amount.
        </p>
      </div>
      <div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-8">
        {filtered.map(v => (
          <VehicleCard key={v.id} vehicle={v} onBook={(veh) => onBook(veh,
'SHOWROOM_PREBOOK')} isLoggedIn={!user} />
        ))}
      </div>
    </div>
  );
};

export default Showroom;

```

AI TRIP PLANNER:

```

import React, { useState } from 'react';
import { generateTripAdvice } from '../services/geminiService';
import { Sparkles, Send, Bot, User as UserIcon, Loader2 } from 'lucide-react';
const TripPlannerAI: React.FC = () => {
  const [query, setQuery] = useState("");

```

```

const [response, setResponse] = useState<string | null>(null);
const [loading, setLoading] = useState(false);
const handleSearch = async (e: React.FormEvent) => {
  e.preventDefault();
  if (!query.trim()) return;
  setLoading(true);
  setResponse(null);
  const result = await generateTripAdvice(query);
  setResponse(result);
  setLoading(false);
};
return (
  <div className="max-w-4xl mx-auto py-12 px-4">
    <div className="text-center mb-10">
      <div className="inline-flex items-center justify-center p-3 bg-gradient-to-r from-indigo-500 via-purple-500 to-pink-500 rounded-2xl shadow-lg mb-4">
        <Sparkles className="w-8 h-8 text-white" />
      </div>
      <h2 className="text-3xl md:text-4xl font-extrabold text-transparent bg-clip-text bg-gradient-to-r from-indigo-600 to-purple-600 dark:from-indigo-400 dark:to-purple-400 mb-4">
        AI Smart Travel Assistant
      </h2>
      <p className="text-lg text-gray-600 dark:text-gray-300 max-w-2xl mx-auto">
        Not sure where to go or which vehicle fits your trip? Ask our AI assistant powered by Gemini.
      </p>
    </div>

    <div className="bg-white dark:bg-slate-800 rounded-2xl shadow-xl border border-gray-100 dark:border-slate-700 overflow-hidden">
      <div className="p-6 md:p-8">

```

```

<div className="space-y-6 mb-8">
  { /* Chat Display */ }
  { response && (
    <>
      <div className="flex items-start gap-4 justify-end">
        <div className="bg-indigo-100 dark:bg-indigo-900/30 text-indigo-900
dark:text-indigo-100 p-4 rounded-2xl rounded-tr-none max-w-[80%]">
          <p>{ query }</p>
        </div>
        <div className="w-10 h-10 rounded-full bg-indigo-100 dark:bg-indigo-
900 flex items-center justify-center flex-shrink-0">
          <UserIcon className="w-6 h-6 text-indigo-600 dark:text-indigo-400" />
        </div>
      </div>
      <div className="flex items-start gap-4">
        <div className="w-10 h-10 rounded-full bg-purple-100 dark:bg-purple-
900 flex items-center justify-center flex-shrink-0">
          <Bot className="w-6 h-6 text-purple-600 dark:text-purple-400" />
        </div>
        <div className="bg-gray-50 dark:bg-slate-900 p-6 rounded-2xl rounded-
tl-none w-full border border-gray-100 dark:border-slate-700">
          <div className="prose dark:prose-invert max-w-none text-gray-800
dark:text-gray-200 whitespace-pre-line">
            { response }
          </div>
        </div>
      </div>
    </>
  ) }
  { !response && !loading && (
    <div className="flex flex-col items-center justify-center py-12 text-center text-
gray-400 dark:text-gray-500">

```

```

        <Bot className="w-16 h-16 mb-4 opacity-20" />

        <p>Try asking: "Plan a 3-day beach trip for a family of 4" or "Best car for
mountain driving?"</p>

    </div>

    )}

</div>

{ /* Input Area */}

<form onSubmit={handleSearch} className="relative">

    <input

        type="text"

        value={query}

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

        placeholder="Ask for travel advice or itinerary planning..."

        className="w-full p-4 pr-16 rounded-xl border border-gray-200 dark:border-
slate-600 bg-gray-50 dark:bg-slate-900 text-gray-900 dark:text-white focus:ring-2 focus:ring-
indigo-500 focus:border-transparent outline-none transition-all shadow-inner"

        />

    <button

        type="submit"

        disabled={loading || !query.trim()}

        className="absolute right-2 top-2 bottom-2 bg-indigo-600 hover:bg-indigo-700
disabled:bg-gray-400 text-white p-3 rounded-lg transition-colors flex items-center justify-
center aspect-square"

        >

        {loading ? <Loader2 className="w-5 h-5 animate-spin" /> : <Send
className="w-5 h-5" />}

    </button>

</form>

</div>

<div className="bg-gray-50 dark:bg-slate-900/50 px-6 py-3 border-t border-gray-100
dark:border-slate-700 flex justify-between items-center text-xs text-gray-500">

    <span>Powered by Google Gemini 2.5 Flash</span>

    <span>AI responses may vary. Always verify details.</span>

```

```

        </div>

    </div>

</div>

);

};

export default TripPlannerAI;

```

VEHICLE CARD PAGE:

```

import React from 'react';

import { Vehicle, VehicleCategory } from '../types';

import { Users, Fuel, Settings, Star, Zap, Tag } from 'lucide-react';

interface VehicleCardProps {
  vehicle: Vehicle;
  onBook: (vehicle: Vehicle) => void;
  isLoggedIn: boolean; // Passed to handle button text
}

const VehicleCard: React.FC<VehicleCardProps> = ({ vehicle, onBook, isLoggedIn }) => {
  const isShowroom = vehicle.category === VehicleCategory.SHOWROOM;

  return (
    <div className="bg-white dark:bg-slate-800 rounded-xl shadow-lg overflow-hidden
    hover:shadow-xl transition-all duration-300 border border-gray-100 dark:border-slate-700
    flex flex-col h-full group relative">

      <div className="relative h-48 overflow-hidden">

        <img

          src={vehicle.image}

          alt={vehicle.name}

          className="w-full h-full object-cover group-hover:scale-105 transition-transform
          duration-500"

        />

        <div className="absolute top-4 right-4 bg-white/90 dark:bg-slate-900/90 backdrop-
        blur px-2 py-1 rounded-lg shadow-sm flex items-center">

          <Star className="w-3 h-3 text-amber-500 fill-amber-500 mr-1" />

```

```

    <span className="text-xs font-bold text-gray-800 dark:text-gray-
200">{ vehicle.rating }</span>

    </div>

    { /* Badge for Category */ }

    <div className={`absolute top-4 left-4 px-3 py-1 rounded-full text-xs font-bold
uppercase tracking-wider shadow-sm ${
        isShowroom
        ? 'bg-purple-600 text-white'
        : 'bg-indigo-600 text-white'
    }`}>

        { isShowroom ? 'Pre-Order' : 'Rental' }

    </div>

    { !vehicle.available && (

        <div className="absolute inset-0 bg-black/50 flex items-center justify-center">

            <span className="bg-red-500 text-white px-3 py-1 rounded-full text-sm font-
medium">Sold Out</span>

        </div>

    ) }

</div>

<div className="p-5 flex flex-col flex-grow">

    <div className="flex justify-between items-start mb-2">

        <div>

            <span className="text-xs font-semibold text-indigo-600 dark:text-indigo-400
uppercase tracking-wide">{ vehicle.type }</span>

            <h3 className="text-lg font-bold text-gray-900 dark:text-white leading-
tight">{ vehicle.name }</h3>

        </div>

        </div>

        <p className="text-gray-500 dark:text-gray-400 text-sm mb-4 line-clamp-
2">{ vehicle.description }</p>

        <div className="flex items-center space-x-4 mb-6 text-sm text-gray-600 dark:text-
gray-300">

            <div className="flex items-center" title="Capacity">

```



```

        <Users className="w-4 h-4 mr-1.5 opacity-70" />
        <span>{ vehicle.capacity}</span>
    </div>

    <div className="flex items-center" title="Transmission">
        <Settings className="w-4 h-4 mr-1.5 opacity-70" />
        <span>{ vehicle.transmission === 'Automatic' ? 'Auto' : 'Manual'}</span>
    </div>

    <div className="flex items-center" title="Fuel Type">
        { vehicle.fuel === 'Electric' ? <Zap className="w-4 h-4 mr-1.5 text-yellow-500" /> :
    <Fuel className="w-4 h-4 mr-1.5 opacity-70" />}
        <span>{ vehicle.fuel}</span>
    </div>
</div>

{isShowroom && vehicle.fullPrice && (
    <div className="mb-4 bg-gray-50 dark:bg-slate-900/50 p-2 rounded text-xs text-gray-500 flex items-center">
        <Tag className="w-3 h-3 mr-1" />
        Ex-Showroom: <span className="font-bold ml-1 text-gray-700 dark:text-gray-300">${ vehicle.fullPrice.toLocaleString()}</span>
    </div>
)}

<div className="mt-auto flex items-center justify-between pt-4 border-t border-gray-100 dark:border-slate-700">
    <div>
        {isShowroom ? (
            <div>
                <span className="text-2xl font-bold text-gray-900 dark:text-white">${ vehicle.bookingFee}</span>
                <span className="text-gray-500 dark:text-gray-400 text-xs block">Booking Token</span>
            </div>
        ) : (

```

```

    </>
    <span className="text-2xl font-bold text-gray-900 dark:text-
white">${vehicle.pricePerDay}</span>
    <span className="text-gray-500 dark:text-gray-400 text-sm"> /day</span>
  </>
)}
</div>
<button
  onClick={() => onBook(vehicle)}
  disabled={!vehicle.available}
  className={`px-4 py-2 rounded-lg font-medium transition-colors shadow-lg ${
    vehicle.available
      ? isShowroom
        ? 'bg-purple-600 hover:bg-purple-700 text-white shadow-purple-500/20'
        : 'bg-indigo-600 hover:bg-indigo-700 text-white shadow-indigo-500/20'
        : 'bg-gray-200 dark:bg-slate-700 text-gray-400 cursor-not-allowed shadow-none'
      }`
  >
  {!vehicle.available
    ? 'Unavailable'
    : !isLoggedIn
      ? 'Login'
      : isShowroom ? 'Pre-Book' : 'Rent Now'}
  </button>
</div>
</div>
);
};
export default VehicleCard;

```

APP DETAILS:

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

import Layout from './components/Layout';

import AdminDashboard from './components/AdminDashboard';

import TripPlannerAI from './components/TripPlannerAI';

import Auth from './components/Auth';

import Rentals from './components/Rentals';

import Showroom from './components/Showroom';

import Packages from './components/Packages';

import MyBookings from './components/MyBookings';

import Home from './components/Home';

import { User, UserRole, Booking } from './types';

import { CheckCircle, CreditCard, Smartphone, Globe, Calendar, AlertCircle } from 'lucide-react';

type UserCredential = {
  user: User;
  password: string;
};

const App: React.FC = () => {
  const [darkMode, setDarkMode] = useState(false);
  const [currentView, setCurrentView] = useState('auth');
  const [user, setUser] = useState<User | null>(null);
  // Persistent User Database (Email -> Credential)
  const [userDb, setUserDb] = useState<Map<string, UserCredential>>(new Map());
  const [bookings, setBookings] = useState<Booking[]>([]);
  const [showBookingModal, setShowBookingModal] = useState(false);
  const [selectedItem, setSelectedItem] = useState<any>(null);
  const [pendingBookingItem, setPendingBookingItem] = useState<any>(null);
  // Payment State
  const [paymentMethod, setPaymentMethod] = useState<'CARD' | 'UPI' | 'NET_BANKING'>('CARD');
```

```

// Date Selection State for Rentals

const [startDate, setStartDate] = useState("");
const [endDate, setEndDate] = useState("");
const [totalDays, setTotalDays] = useState(1);
useEffect(() => {
  if (darkMode) {
    document.documentElement.classList.add('dark');
  } else {
    document.documentElement.classList.remove('dark');
  }
}, [darkMode]);

// Auth Logic for Strict Enforcement

const registerUser = (newUser: User, password: string): boolean => {
  if (userDb.has(newUser.email)) {
    return false; // User already exists
  }
  const newDb = new Map(userDb);
  newDb.set(newUser.email, { user: newUser, password });
  setUserDb(newDb);
  return true;
};

const authenticateUser = (email: string, password: string): User | null => {
  const credential = userDb.get(email);
  if (credential && credential.password === password) {
    return credential.user;
  }
  return null;
};

const handleLogin = (userData: User) => {

```

```

setUser(userData);
if (pendingBookingItem) {
  setSelectedItem(pendingBookingItem);
  // Initialize dates if it's a vehicle
  if (pendingBookingItem.type === 'VEHICLE') {
    const today = new Date().toISOString().split('T')[0];
    const tomorrow = new Date(Date.now() + 86400000).toISOString().split('T')[0];
    setStartDate(today);
    setEndDate(tomorrow);
    setTotalDays(1);
  }
  setShowBookingModal(true);
  setPendingBookingItem(null);
  setCurrentView('vehicles');
} else {
  setCurrentView('home');
}
};

```

```

const handleLogout = () => {
  setUser(null);
  setCurrentView('auth');
  setPendingBookingItem(null);
};

```

```

const handleBook = (item: any, type: 'VEHICLE' | 'PACKAGE' |
'SHOWROOM_PREBOOK') => {
  const itemWithBookingType = { ...item, type };
  if (!user) {
    const confirmLogin = window.confirm("You must be logged in to book. Go to login
page?");
    if (confirmLogin) {

```

```

        setPendingBookingItem(itemWithBookingType);
        setCurrentView('auth');
    }
    return;
}

// Set default dates for rentals
if (type === 'VEHICLE') {
    const today = new Date().toISOString().split('T')[0];
    const tomorrow = new Date(Date.now() + 86400000).toISOString().split('T')[0];
    setStartDate(today);
    setEndDate(tomorrow);
    setTotalDays(1);
}

setSelectedItem(itemWithBookingType);
setShowBookingModal(true);
};

// Recalculate days when dates change
useEffect(() => {
    if (startDate && endDate) {
        const start = new Date(startDate);
        const end = new Date(endDate);
        const diffTime = Math.abs(end.getTime() - start.getTime());
        const diffDays = Math.ceil(diffTime / (1000 * 60 * 60 * 24));
        // If end date is same as start date, count as 1 day
        setTotalDays(diffDays === 0 ? 1 : diffDays);
    }
}, [startDate, endDate]);

const confirmBooking = () => {
    if (!selectedItem || !user) return;
    // Validation for Rentals

```

```

if (selectedItem.type === 'VEHICLE') {
  if (!startDate || !endDate) {
    alert("Please select both Start Date and End Date.");
    return;
  }
  if (new Date(endDate) < new Date(startDate)) {
    alert("End Date cannot be before Start Date.");
    return;
  }
}

let finalPrice = 0;

if (selectedItem.type === 'VEHICLE') {
  // Calculate based on days
  finalPrice = selectedItem.pricePerDay * totalDays;
}

else if (selectedItem.type === 'SHOWROOM_PREBOOK') {
  finalPrice = selectedItem.bookingFee;
}

else {
  finalPrice = selectedItem.price;
}

const newBooking: Booking = {
  id: `bk_${Date.now()}`,
  itemId: selectedItem.id,
  itemType: selectedItem.type,
  itemName: selectedItem.name || selectedItem.title,
  date: new Date().toLocaleDateString(), // Transaction date
  startDate: selectedItem.type === 'VEHICLE' ? startDate : undefined,
  endDate: selectedItem.type === 'VEHICLE' ? endDate : undefined,
  status: 'Confirmed',

```

```

    totalPrice: finalPrice,
    customerName: user.name
  };
  setBookings(prev => [newBooking, ...prev]);
  setShowBookingModal(false);
  setSelectedItem(null);

  alert(`Success! Payment via ${paymentMethod} received. Booking Confirmed for
  ${finalPrice}.`);

  if(user.role === UserRole.ADMIN) {
    setCurrentView('admin');
  }
};

const cancelBooking = (bookingId: string) => {
  const bookingToCancel = bookings.find(b => b.id === bookingId);
  if (!bookingToCancel) return;

  if (window.confirm(`Are you sure you want to cancel your booking for
  ${bookingToCancel.itemName}?`)) {
    setBookings(prevBookings =>
      prevBookings.map(b =>
        b.id === bookingId ? { ...b, status: 'Cancelled' } : b
      )
    );
  }
};

if (currentView === 'auth') {
  return (
    <div className={`fixed top-4 right-4 z-50`}>
      <button onClick={() => setDarkMode(!darkMode)} className="p-2 bg-white
      dark:bg-slate-800 rounded-full shadow-lg text-xl">
        {darkMode ? '☀️' : '🌙'}
      </button>
    </div>
  );
}

```



```

        </button>

    </div>

    <Auth
        onLogin={handleLogin}
        onRegister={registerUser}
        onAuthenticate={authenticateUser}
    />

</>

);
}

// Calculate dynamic price for display in modal
const getDynamicPrice = () => {
    if (!selectedItem) return 0;
    if (selectedItem.type === 'VEHICLE') {
        return (selectedItem.pricePerDay || 0) * totalDays;
    }
    if (selectedItem.type === 'SHOWROOM_PREBOOK') {
        return selectedItem.bookingFee || 0;
    }
    return selectedItem.price || 0;
}

return (
    <
        <Layout
            darkMode={darkMode}
            toggleDarkMode={() => setDarkMode(!darkMode)}
            currentUser={user}
            onLogin={() => setCurrentView('auth')}
            onLogout={handleLogout}
            currentView={currentView}

```

```

        setView={setCurrentView}
    >
    {currentView === 'home' && <Home user={user} setView={setCurrentView}
onBook={handleBook} />}
    {currentView === 'vehicles' && <Rentals user={user} onBook={handleBook}
setView={setCurrentView} />}
    {currentView === 'showroom' && <Showroom user={user} onBook={handleBook}
/>}
    {currentView === 'packages' && <Packages user={user} onBook={handleBook} />}
    {currentView === 'ai-planner' && <TripPlannerAI />}
    {currentView === 'admin' && user?.role === UserRole.ADMIN && (
        <div className="max-w-7xl mx-auto px-4 py-8">
            <AdminDashboard bookings={bookings} />
        </div>
    )}
    {currentView === 'admin' && user?.role !== UserRole.ADMIN && <MyBookings
user={user} bookings={bookings} onCancel={cancelBooking} setView={setCurrentView}
/>}
    {currentView === 'my-bookings' && <MyBookings user={user}
bookings={bookings} onCancel={cancelBooking} setView={setCurrentView} />}
</Layout>
{ /* Booking & Payment Modal */}
{showBookingModal && selectedItem && (
    <div className="fixed inset-0 z-[100] flex items-center justify-center bg-black/60
backdrop-blur-sm p-4 animate-fade-in">
        <div className="bg-white dark:bg-slate-800 rounded-2xl w-full max-w-md p-6
shadow-2xl transform transition-all scale-100 max-h-[90vh] overflow-y-auto">
            <div className="text-center mb-6">
                <div className={`w-12 h-12 rounded-full flex items-center justify-center
mx-auto mb-4 ${
                    selectedItem.type === 'SHOWROOM_PREBOOK'
                        ? 'bg-purple-100 dark:bg-purple-900/30'
                        : 'bg-green-100 dark:bg-green-900/30'
                }`}>

```

```

        {selectedItem.type === 'SHOWROOM_PREBOOK'
          ? <CheckCircle className="w-6 h-6 text-purple-600" />
          : <CheckCircle className="w-6 h-6 text-green-600" />
        }
      </div>

      <h3 className="text-2xl font-bold text-gray-900 dark:text-white">
        {selectedItem.type === 'SHOWROOM_PREBOOK' ? 'Confirm Pre-
Booking' : 'Complete Your Booking'}
      </h3>
    </div>

    <div className="bg-gray-50 dark:bg-slate-900 p-4 rounded-xl mb-6 space-y-3
text-sm border border-gray-100 dark:border-slate-700">
      <div className="flex justify-between">
        <span className="text-gray-500">Item:</span>
        <span className="font-bold text-gray-900 dark:text-white text-
right">{selectedItem.name || selectedItem.title}</span>
      </div>

      { /* RENTAL SPECIFIC INPUTS */ }
      {selectedItem.type === 'VEHICLE' && (
        <div className="pt-2 border-t border-gray-200 dark:border-slate-700 mt-2
space-y-3">
          <div className="grid grid-cols-2 gap-3">
            <div>
              <label className="text-xs text-gray-500 block mb-
1">From</label>
              <input
                type="date"
                value={startDate}
                min={new Date().toISOString().split('T')[0]}
                onChange={(e) => setStartDate(e.target.value)}
                className="w-full p-2 bg-white dark:bg-slate-800 border border-
gray-200 dark:border-slate-700 rounded-lg text-xs"

```

```

        />
    </div>
    <div>
        <label className="text-xs text-gray-500 block mb-1">To</label>
        <input
            type="date"
            value={endDate}
            min={startDate}
            onChange={(e) => setEndDate(e.target.value)}
            className="w-full p-2 bg-white dark:bg-slate-800 border border-
gray-200 dark:border-slate-700 rounded-lg text-xs"
        />
    </div>
</div>
<div className="flex justify-between items-center bg-blue-50 dark:bg-
blue-900/20 p-2 rounded text-blue-800 dark:text-blue-300">
    <span className="text-xs font-medium flex items-center"><Calendar
className="w-3 h-3 mr-1" /> Duration</span>
    <span className="text-xs font-bold">{totalDays} Day{totalDays > 1
? 's' : ''}</span>
</div>
<div className="flex justify-between text-xs text-gray-500">
    <span>Rate</span>
    <span>${selectedItem.pricePerDay} / day</span>
</div>
</div>
))
<div className="flex justify-between pt-2 border-t border-gray-200
dark:border-slate-700">
    <span className="text-gray-500 font-medium">
        Total Payable:
    </span>

```

```

        <span className="font-bold text-indigo-600 dark:text-indigo-400 text-xl">
            ${getDynamicPrice()}
        </span>
    </div>
</div>
{ /* Payment Options */ }
<div className="mb-6">
    <h4 className="text-sm font-semibold text-gray-700 dark:text-gray-300 mb-3">Select Payment Method</h4>
    <div className="space-y-2">
        <label className={`flex items-center p-3 rounded-lg border cursor-pointer transition-colors ${paymentMethod === 'CARD' ? 'border-indigo-500 bg-indigo-50 dark:bg-indigo-900/20' : 'border-gray-200 dark:border-slate-700'} `}>
            <input type="radio" name="payment" className="mr-3"
checked={paymentMethod === 'CARD'} onChange={() => setPaymentMethod('CARD')} />
            <CreditCard className="w-4 h-4 mr-2 text-gray-500" />
            <span className="text-sm text-gray-700 dark:text-gray-300">Credit /
Debit Card</span>
        </label>
        <label className={`flex items-center p-3 rounded-lg border cursor-pointer transition-colors ${paymentMethod === 'UPI' ? 'border-indigo-500 bg-indigo-50 dark:bg-indigo-900/20' : 'border-gray-200 dark:border-slate-700'} `}>
            <input type="radio" name="payment" className="mr-3"
checked={paymentMethod === 'UPI'} onChange={() => setPaymentMethod('UPI')} />
            <Smartphone className="w-4 h-4 mr-2 text-gray-500" />
            <span className="text-sm text-gray-700 dark:text-gray-300">UPI /
GPay / PhonePe</span>
        </label>
        <label className={`flex items-center p-3 rounded-lg border cursor-pointer transition-colors ${paymentMethod === 'NET_BANKING' ? 'border-indigo-500 bg-indigo-50 dark:bg-indigo-900/20' : 'border-gray-200 dark:border-slate-700'} `}>
            <input type="radio" name="payment" className="mr-3"
checked={paymentMethod === 'NET_BANKING'} onChange={() =>
setPaymentMethod('NET_BANKING')} />
            <Globe className="w-4 h-4 mr-2 text-gray-500" />

```

```

        <span className="text-sm text-gray-700 dark:text-gray-300">Net
Banking</span>

        </label>

    </div>

</div>

<div className="flex gap-3">

    <button

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

        className="flex-1 px-4 py-3 bg-gray-100 hover:bg-gray-200 dark:bg-slate-
700 dark:hover:bg-slate-600 text-gray-700 dark:text-gray-300 rounded-xl font-medium
transition-colors"

        >

            Cancel

        </button>

        <button

            onClick={confirmBooking}

            className={`flex-1 px-4 py-3 text-white rounded-xl font-medium
transition-colors shadow-lg ${

                selectedItem.type === 'SHOWROOM_PREBOOK'

                ? 'bg-purple-600 hover:bg-purple-700 shadow-purple-500/30'

                : 'bg-indigo-600 hover:bg-indigo-700 shadow-indigo-500/30'

            }`}

            >

                Pay ${getDynamicPrice()}

            </button>

        </div>

    </div>

</div>

    )}

</>

);

};

```

```
export default App;
```

INDEX CONFIGURATION:

```
import React from 'react';
```

```
import ReactDOM from 'react-dom/client';
```

```
import App from './App';
```

```
const rootElement = document.getElementById('root');
```

```
if (!rootElement) {
```

```
  throw new Error("Could not find root element to mount to");
```

```
}
```

```
const root = ReactDOM.createRoot(rootElement);
```

```
root.render(  
  <React.StrictMode>
```

```
    <App />
```

```
  </React.StrictMode>
```